FREE ENTRY COMPETITION SOFTWARE TO BE WON

ZX COMPUTING

FEB/MAR 83
£1.85

MACHINE CODE
PT 3
94

Britain's Biggest Magazine For The Sinclair User

Over 120 pages of information and programs for the ZX Spectrum, ZX81 and ZX80 Computers

PLUS

- Software reviews – are you getting value for money?
- A complete education package for younger users
- Getting down to business on your ZX81
- Become a machine code master
- Games galore for you to RUN

SPECTRUM COMES TO LIFE !

ABBEX

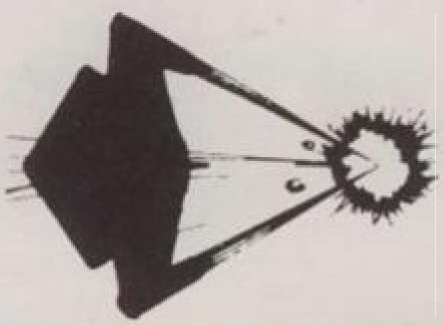
THE GAMES PEOPLE

SPOOKYMAN



EAT THE PILLS AND KEEP AWAY FROM THE GHOSTS. EAT THE VITAMINS AND CHASE THE GHOSTS. FULL MACHINE CODE ARCADE ACTION AND SOUND. DESIGNED TO MAKE YOUR ADRENALINE RUN HIGH. PRICE £4.95

COSMOS



YOU HAVE THE JOB OF PROTECTING A CONVOY FROM ALIENS AND METEOR STORMS. WATCH OUT FOR THE SPACE MINES AS YOU PLAY THIS AMAZING MACHINE CODE ARCADE ACTION SIMULATION PROGRAM. PRICE £4.95

OTHER GAMES IN OUR SPECTRUM RANGE INCLUDE: ANDROIDS RUN, HIGH NOON, ETX, INVASION, FAUST'S FOLLY (Adventure), HARRIER - ALL AT £5.95

ABBEX ELECTRONICS LTD.

20 ASHLEY COURT, GREAT NORTHWAY, LONDON NW4
Tel: 01-203 1465

PLEASE COMPLETE IN BLOCK CAPS
NAME (Mr/Mrs/Miss) ....
ADDRESS ....
(CARD HOLDERS) ....
POST CODE .... PHONE NO. ....

Make cheques payable to ABBEX ELECTRONICS LTD.

I enclose my cheque for £

I wish to pay by ☐ CHEQUE ☐ ACCESS
Please debit my creditcard a/c
NUMBER ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Signed ....
date ....

Please send me the following
(Allow 14 days for delivery)

☐ **SPOOKYMAN** PRICE £4.95

☐ **COSMOS** PRICE £4.95

☐ **OTHER GAME** .... PRICE £5.95

☐ **CATALOGUE**

OTHER-WORLDLY SOFTWARE SOFTEK AT DOWN TO EARTH PRICES

Imagine a program which instantly converts SPECTRUM BASIC into Machine Code, and transforms the SPECTRUM into a micro up to 100 times faster than a BBC Model B...

48K ZX SPECTRUM
COMPILER

METEOROIDS

This version has been hailed the best: 3 sizes of meteoroids, saucer, wrap-around screen, bonus ship at 10000 points, amazing explosions, rotate left/right, thrust, fire, hyperspace and shield. Full feature 'arcade' sounds and full colour. Works with Joysticks (from us at £19.50). Only £4.95 inc (16K or 48K).

SOFTIME

Amazing piece of machine code programming - produces a continuous real time clock on screen all the time - even when running/writing another program! Has alarm facility. For 16K or 48K, only £3.95.

SOFTEK SPECTRUM SOFTWARE

COMPILER
Benchmarks

	BBC	SPECTRUM (BASIC)	SPECTRUM (COMPILER)
BM1	1.0s	4.8	0.15
BM2	3.1	8.7	0.09
BM3	8.2	21.1	1.10
BM4	8.7	20.4	0.99
BM5	9.1	24.0	1.11
BM6	13.9	55.3	1.91
BM7	21.4	80.7	2.14

What the SUPER C compiler does is almost magical. It turns a program written in Spectrum BASIC into a machine code equivalent. SUPER C is child's play to use - simply type in a program as usual (or load from tape) and in a few key presses SUPER C turns it instantly into a super fast machine code version. It will compile virtually any Spectrum BASIC statement, plus a few features not available to you usually, but extra thought will be required for strings and floating point. There is no program you can write in BASIC which cannot be turned into a SUPER C version. However, enhancement packages will be available in the New Year to aid string handling, floating point, and will offer a host of other features - each for a nominal cost. With SUPER C your Spectrum will run faster than with either FORTH or PASCAL - and no new language to learn! SUPER C comes complete with extensive documentation, a manual written by Tim Langdell, example programs, and all boxed in a professional library case. Less than half the price of a compiler for any comparable micro, - £14.95 inc.

48K 3D MAZENTURE

A maze program and adventure in one. Amazing 'monster chase' animation and more than a hint of dungeons and dragons. 48K only, £5.95.

THE ZOLAN ADVENTURE

A full adventure crammed into 16K! Quite astounding how extensive and engaging this program is. £4.95.

Please rush me . . . .

Mr/Mrs . . . .
Address . . . .

or for INSTANT CREDIT CARD order 7 days a week phone: 01-930 9232.

Send cheques or POs to: 329 Croxted Road, London SE24. Tel: 01-674 4572

SOFTEK, SPECTRUM SOFTWARE

NOW AVAILABLE FROM HILTON COMPUTER SERVICES
(Author of the Best Banking Programs in the World)

THE PBS BANK RECONCILIATION IS BELIEVED TO BE THE FIRST SUCH PROGRAM TO BE AVAILABLE FOR THE SINCLAIR ZX81 AND ZX SPECTRUM

Just look at some of its features:

- Automatically matches bank statement items with transactions recorded in your **PERSONAL BANKING SYSTEM**.
- Lists all outstanding items such as uncleared cheques, funds received but not advised to you, bank charges — highlights all discrepancies. No more 'scratching around' to try and find out why your statement balance is different to what you expected.
- Full search facility, allowing on-screen correction of any item.
- Saves and loads data to and from tape (ZX81 version at double speed). Exchanges data with the PBS program. No need to save programs.
- Single key operation — uses machine code keyboard scan for single letter commands (ZX81 version).
- Extremely easy to use. No jargon to wade through before you get started. Demonstration program included.
- All item lists, searches (with totals). Can be output to your ZX Printer.
- Full instructions included.
- Unique after sale maintenance provided.
- Operates in conjunction with the 'Personal Banking System' (full input & correction facilities plus automatic posting of standing orders on due dates and fully detailed statements).

So banish the headaches, make life easier for yourself, send £15 for the **PERSONAL BANKING SYSTEM + PBS BANK RECONCILIATION** and users manual.

Existing clients need only send £5 for the **PBS BANK RECONCILIATION** which will operate with their present PBS.

THE PERSONAL BANKING SYSTEM (as reviewed in Popular Computing Weekly — 1st July) is available separately for £9.95 for cassette & user manual. ZX81 PBS owners can order the Spectrum version for just £5.

Please specify whether ZX81 (16K) or ZX Spectrum (48K)

GUARANTEED SOFTWARE AVAILABLE FROM

HILTON COMPUTER SERVICES,
14 Avalon Road, Oprington, Kent BR6 9AX
YOUR PBS IS NEVER OUT OF DATE!

JOYSTICKS



23.50 INC VAT & P+P

ZX Spectrum / ZX81

FOR ONE JOYSTICK AND INTERFACE MODULE

BUILT, TESTED & READY FOR USE

- ★ **NO SOLDERING,** interface module plugs into rear connector between ZX and Ram Pack, Printer etc.
- ★ **NO SPECIAL PROGRAMMING,** Joystick 1 simulates cursor movement keys 5 to 8 through interface module.
- ★ **IMMEDIATELY COMPATIBLE WITH ALL SOFTWARE** using the arrow keys for movement.
- ★ **TWO JOYSTICKS** connect via one interface module.
- ★ **EIGHT DIRECTION, SELF CENTRALISING ACTION** with integral 'FIRE' button.
- ★ **FREE 'VIDEO GRAFFITI'** program & full instructions.

IMPORTANT: Use will not invalidate your Sinclair guarantee.

FROM: MR/MRS/MISS

ADDRESS

SEND C.W.O. To: A.G.F. HARDWARE, Dept. SP, 26 Van Gogh Place, Bognor Regis, West Sussex PO22 9BY

QTY	ITEM	ITEM PRICE	TOTAL
	JOYSTICK	7.54	
	INTERFACE MODULE	15.96	
ZX Spectrum ☐ ZX81 ☐ Please tick		FINAL TOTAL	

ZX COM

ZX Computing
Volume One
Number Five
Feb/March 1983

Deputy Editor: Roger Munford
Advertising Manager: Jeff Raggett
Divisional Advertising Manager: Beverley McNeill
Managing Editor: Ron Harris
Managing Director: T J Connell

Origination and design by MM Design & Print, 145 Charing Cross Road, London WC2H 0EE.

Published by Argus Specialist Publications Ltd, 145 Charing Cross Road, London WC2H 0EE.

CONTENTS

MPUTING

ZX Computing is published bi-monthly on the fourth Friday of the month. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. 01-247 8233. Printed by: Henry Garnett Ltd., Rotherham.

ZX Computing is constantly on the look-out for well-written articles and programs. If you think that your efforts meet our standards, please feel free to submit your work to us for consideration.

All submitted material should be typed if possible; handwritten work will be considered, but please use your neatest handwriting. Any programs submitted should be listed, a cassette of your program alone will not be considered. All programs must come complete with a full explanation of the operation and, where relevant, the structure; Spectrum programs should be accompanied with a cassette of the program (which will be returned) as well as the listing.

All submissions will be acknowledged and any published work will be paid for at competitive rates. All work for consideration should be sent to the Deputy Editor at our Charing Cross Road address.

NEW! ZX SPECTRUM TAPE NOW READY!
NEW! EXPANDED DISC VERSIONS FOR APPLE, PET AND SHARP!

NOW AVAILABLE FOR THE ZX SPECTRUM!

THE VALLEY



What are you... Barbarian or Wizard?

Choose your character type carefully... Barbarians recover quickly but their magic doesn't come easily. A Wizard? Slow on the draw and slow to mature...but live long enough and grow wise enough and your lightning bolts are almost unstoppable...

The Valley is a real-time game of adventure and survival. You may choose one of five character types to be your personal 'extension of self' to battle and pit your wits against a number of monsters. Find treasure, fight a Thunder-Lizard in the arid deserts of the Valley, conquer a Kraken in the lakes surrounding the dread Temples of Y'Nagioth or cauterise a Wraith in the Black Tower. In fact, live out the fantasies you've only dared dream about. BUT BEWARE... more die than live to tell the tale.

The Valley runs in a 48K Spectrum (a 16K version will be available shortly) and makes full usage of the excellent colour and graphics capability of the machine. As you battle your way up the ratings your character can be saved onto tape, to re-enter the Valley another day — if you dare!

Full instructions are included with the game, but if you want more detail on the program, a 16 page reprint of the orignal 'Computing Today' article is available at £1.95 all inc.

If you have the courage, the Valley costs only £11.45 all inc. from ASP SOFTWARE, 145 Charing Cross Road, London WC2 0EE.

Please send me...tape(s) of The Valley at £11.45 per tape all inc.
and..........listings at £1.95 each.
I enclose my cheque/Postal Order/International Money Order for: (delete as necessary)
£.....(Made payable to ASP Ltd)
or Debit my Access/Barclaycard
(delete as necessary)

Please use BLOCK CAPITALS
NAME(Mr/Mrs/Miss) ....

ADDRESS ....

.... POSTCODE ....

Signature .... Date ....

ZX2

ASP SOFTWARE

Welcome



First of all, I hope you all had a nice time over the Christmas holidays, and welcome to those of you who received a little 'computing something' from Santa Claus and are eager to flex your programming skills in the absorbing hobby of home computing. Within these pages, you'll find lots to interest you, with features and programs for the complete range of Sinclair computers as well as useful hints and routines for the experienced and inexperienced user alike.

A fresh face

As you may have noticed (if you've taken a PEEK at the name at the end of this column), Tim Hartnell has departed the Editor's chair of ZX Computing. Tim has decided to concentrate on writing books for the time being. After all, now that a certain computing magazine has crowned him the 'Barbara Cartland of the computer book field', Tim feels he has to do something to keep his image shining!

Nevertheless, he has not departed these pages for good, and we will bring you his wise words of wisdom in future issues. We will also try to maintain the very high standard he set for *ZX Computing* in the first four issues.

Indeed, in this issue, we have a wealth of material for the machine code programmer, continuing Toni Baker's splendid series on 'Mastering machine code on your Spectrum' as well as Ian Turtle re-living some of the traumas of writing his first program in machine code. For those of you searching through the wide range of commercially available software, our reviewers have been busy sifting through the latest cassettes to help you make your choice. We also take a brief look at the huge selection of books available for the Spectrum user.

And now for the good news

Thank you to all those of you who write in to us at *ZX Computing*. As you can see from the wide range of correspondence on the following pages, we certainly have some very interesting readers out there. So, keep those letters coming — it's good to hear your views on the magazine, the industry in general and your discoveries on your computer.

Also, I hope you'll notice the Club Corner page. If you would like some free publicity for your users' club then please write and tell us, giving as many details of times of meetings, places, etc. Joining a club can often be the most important move you make to getting the most out of your computer, so check this page out and see if there's a club near you.

As you read through the letters, I hope you'll notice that there is a lack of letters complaining about Spectrum deliveries. I 'phoned Sinclair the other day and was told by a spokesperson that 'there are now no delivery delays for the ZX Spectrum personal computers'. I was also pleased to hear that the backlog of orders were completely cleared by the end of October, 1982, and that all new orders were being fulfilled within the customary 28 days. Good news indeed.

Contributions

We are always on the lookout for good programs and articles for future issues of *ZX Computing*, and where better to look than to our own readers. If, when reading through the magazine, you think you can write programs as well, or better than, our present contributors, then let's hear from you.

All contributions are, of course, paid for at very competitive rates. So, if you've got your eye on a new ZX add-on or you'd just like to supplement your pocket money, get writing! It is vital, though, that all the programs you send us are totally original, and not 'borrowed' or 'adapted' from other magazines or books. (When Tim was sitting in the Editor's chair, he even received 'original' contributions he himself had written for his own books!)

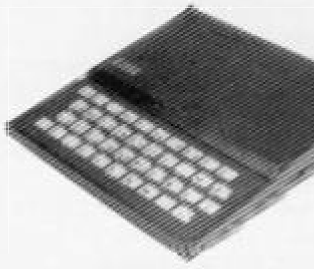
Any kind of program (business, domestic, educational, or just fun) will be welcomed, but particularly those which use ZX BASIC in clever and efficient ways, or those which employ certain routines which can be re-used in other programs.

Program listings are vital, along with a clear explanation of how the program is constructed, what it does and what the user can expect to see once the program is RUN (a screen dump is particularly valuable in this respect). When submitting Spectrum programs, it is very important to remember to enclose a cassette of the program as well as the listing, as this will allow us to check the program before publication.

End byte

I hope you enjoy the contents of this magazine, and that it encourages you to write your own programs to utilise the full potential of your Sinclair machine. But enough of me, I suggest you now get down to the serious business of making the most of your micro with *ZX Computing*.

Roger Munford.



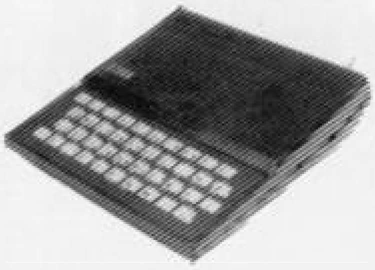
No address

Dear ZX Computing,
You kindly reviewed our 'O' Level Maths software in the August/September issue of *ZX Computing*. However, since then several people have told us of the difficulty in finding our address as it was not given in the article.

As you did mention other people's addresses, we should be glad if you would list our's.
Yours faithfully,

Avril Cowley (Mrs),
Rose Cassettes,
148 Widney Lane,
Solihull,
West Midlands B91 3LH.

● *Sorry about that Avril, these things do happen. I hope that we have more than made up for our omission in the review.*



Thanks a lot!

Dear ZX Computing,
I don't know if it's possible, but through your columns I would like to thank Mr Mike Salem of Hilderbay Ltd, 8/10 Parkway, Regents Park, London NW1 7AA.

I was having problems loading tapes on my ZX81 and when I tried a financial package from Hilderbay had the same difficulty. Mike Salem checked out the tape I returned and reported back to me and sent me another tape by return of post with tips on loading. Needless to say I have had no more trouble.

In fact, I now have a copy of Hilderbay's book entitled *The Microcomputer Users' Book of Tape Recording* which I have found very comprehensive on all aspects of recording, tape recorder and cassette care, loading problems, etc, etc. I think this book has only recently been published.

I very rarely write to magazines, but in this case I feel that I would like to pass on the name of a company offering good service and a willingness to help the 'amateur'. I am in no way related or in fact have never met Mr Salem or anyone from Hilderbay.
Yours faithfully,

JG Marshall,
Sutton Coldfield,
West Midlands.



Any beginners out there?

Dear ZX Computing,
As a recent initiate into the fascinating world of home computing through the purchase of a ZX81, I was so intrigued to see a magazine devoted entirely to ZX computing, I bought issue number 3. In many ways it was a revelation. I had not realized just how many plug-in accessories were now available nor thought much about some of the things which could be done with a ZX computer. It certainly increased my thirst to progress further in the art of programming!

However, in some ways your magazine was also a disappointment. I appreciate your desire to maintain a high standard for published programs, but this need not imply that only long programs are worthy. Yet issue 3 contained very few programs suitable for the standard 1K version of the ZX81, and most of those that were suitable utilised machine code. Beginners (ie would-be experts?), on finding problems with, or wanting to improve upon, such programs would be faced with great difficulty because of a lack of understanding of machine code. Thus, I would appeal to you to put in more material aimed at the beginner. Good, short programs in BASIC, perhaps with a brief explanation of the logic used in them, would be much appreciated I am sure. And the space would not be wasted; after all, today's 1K ZX81 enthusiast is tomorrow's 48K all-singing, all-dancing Spectrum owner!

Perhaps by way of illustration, I could explain a problem which occupied me for several hours recently. I am still working my way through the ZX81 BASIC programming manual which comes with the computer and have reached Chapter 18. On page 121 is a subroutine for drawing an almost straight line from a pixel (A,B) to a pixel (C,D), for use with a program which would generate numerical values of the co-ordinates A,B,C and D. For the purpose of what will follow, imagine the subroutine to be prefaced by a simple set of INPUT statements, as below:

```
10 INPUT A
20 INPUT B
30 INPUT C
40 INPUT D
50 GOSUB 1000
60 STOP
```

Excluding REM statements, the subroutine given in the manual is 26 statements long, and the logic is fairly difficult to follow. I could not avoid feeling that there must be an easier way of programming the drawing of a line which utilises the slope of the line. Indeed, the program represented by lines 1000, 1010, 1030-50, and 1100 (see below) draws a perfectly good line providing the slope (Y/X) is less than or equal to 1. However, for slopes greater than 1, gaps appear in the line since only one pixel is plotted for each value of N. When X=0 (ie a vertical line) the computation breaks down completely because the slope is infinity. The problem of X=0 can easily be overcome by a statement such as:

```
IF X=0 THEN LET X=.5
```

Unfortunately, all of my attempts to write a FOR. . . NEXT loop for a variable Q (of value 0 to the slope of the line), which would allow extra pixels to be plotted at any given value of N, resulted in complications. However, the solution of the problem was really very simple: if the slope of the line when viewed from the X axis is greater than 1 (ie Y/X>1), then the line viewed from the Y direction has a slope less than (ie X/Y<1). Use of this fact allows a very simple subroutine to be written, as follows:

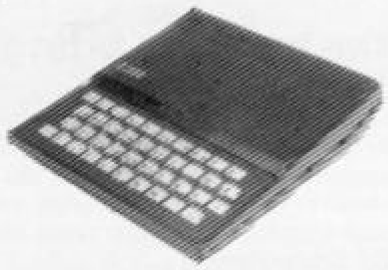
```
1000 LET X=C-A
1010 LET Y=D-B
1020 IF ABS Y >=ABS X
     THEN GOTO 1070
1030 FOR N=A TO C STEP
     SGN X
1040 PLOT N, B+Y/X *
     (N-A)
1050 NEXT N
1060 GOTO 1100
1070 FOR N=B TO D STEP
     SGN Y
1080 PLOT A+X/Y *
     (N-B),N
1090 NEXT N
1100 RETURN
```

The result is a subroutine which involves only 11 lines, and which is conceptually simpler than that in the ZX81 manual. It also uses less memory, so that when used with the preface given above, a complete diagonal line from (0,0) to (63,43) or from (63,0) to (0,43) can still be drawn with the 1K version. This is not the case with the 'official' program.

I am sure this idea will not shatter the world of relatively experienced programmers, but beginners who are puzzled by the subroutine in the manual may find it useful.
Yours faithfully,

K Smith,
Mayols,
Swansea.

● *I appreciate your request for more information and programs for the beginner. In each issue, every effort is made to produce a balance between matrial for both the experienced and inexperienced. I hope that when you read through this magazine, you will find something to interest you.*



What do you think?

Dear ZX Computing,
While investigating a bug in a program of mine, I typed in and ran the following program.

```
10 LET A=256
20 LET A=A/2
30 PRINT,A
40 IF A=0.5 THEN STOP
50 GOTO 20
```

This printed a series of numbers down the right-hand side of the screen, halving each time. However, when it got to 0.5 it did not stop as line 40 commands. Upon changing the number 0.5 to 1, it duly stopped as requested.

I also tried VAL "0.5" in the place of 0.5, but this too went wrong. Is it my programming ineptitude, or have I stumbled across another bug in the ROM.
Yours faithfully,

M Clayton,
Knaresborough,
North Yorkshire.

● *I think the 'bug' you have found is simply due to ZX81's division routines introducing a*

slight error in the continuous division you have asked it to do. Remember that if A is not exactly *equal to 0.5 in line 40, the program will not STOP. Try substituting the following line instead.*

*40 IF INT (A *2)=1 THEN STOP*

Hope this has cleared up your problem.





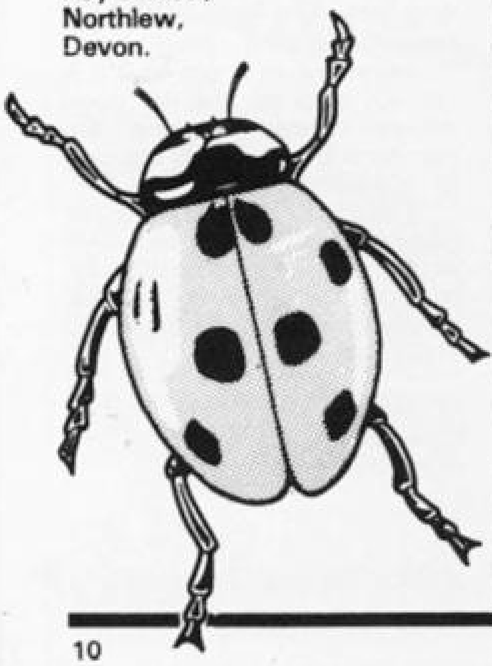
The bugs are biting

Dear ZX Computing,
I would like to bring to your attention an error in the otherwise excellent program Bandit by Messrs. Cleverle and Waring in your October/November issue of ZX Computing.

This concerns the Hex loader at the beginning of the program. Used as listed, it will cause a syntax error on the first input. If line 10 is changed from INPUT X to LET X=16514, all will be well.

I realise this is a small point, but one that could easily 'bug-up' a novice programmer indefinitely.
Yours faithfully,

Roy Barrett,
Northlew,
Devon.



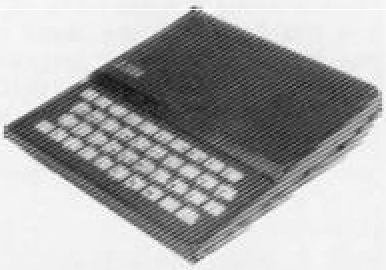
Hold it a minute . . .

Dear ZX Computing,
While writing programs which draw pictures on the screen, I sometimes need to stop a program without showing a report code.

The method I use is to make sure the computer is in Slow mode, then LET L=USR 861. To escape from this routine, you merely press Break.

I hope this will be of use to other ZX81 users.
Yours faithfully,

Alexander Rogers,
Radlett,
Herts.



Two's company

Dear ZX Computing,
I am undergraduate in Computation at the University of Manchester Institute of Science and Technology (UMIST) and have been an owner of Sinclair Research's ZX81 computer since March 1982.

I bought the ZX81 in kit form and it was supplied with a 1.2A PSU which is able to drive the ZX Printer.

I initially wanted the computer for hobbyist purposes, but recently I decided to put it to some commercial use. I thus decided that a printer would be of some use.

When I went to WH Smiths' to have a look at one, I noticed that it was supplied with another 1.2A PSU. I queried this at several branches of WH Smiths and at one, the assistant (whom I presume has had some training) even tried to persuade me that two PSUs were required simultaneously.

I realise that many Sinclair users were provided with a 0.7A PSU with their ZX81s and thus require an updated 1.2A PSU to drive the printer and computer together, but why cannot Sinclair Research provide the updated supply with the computer?

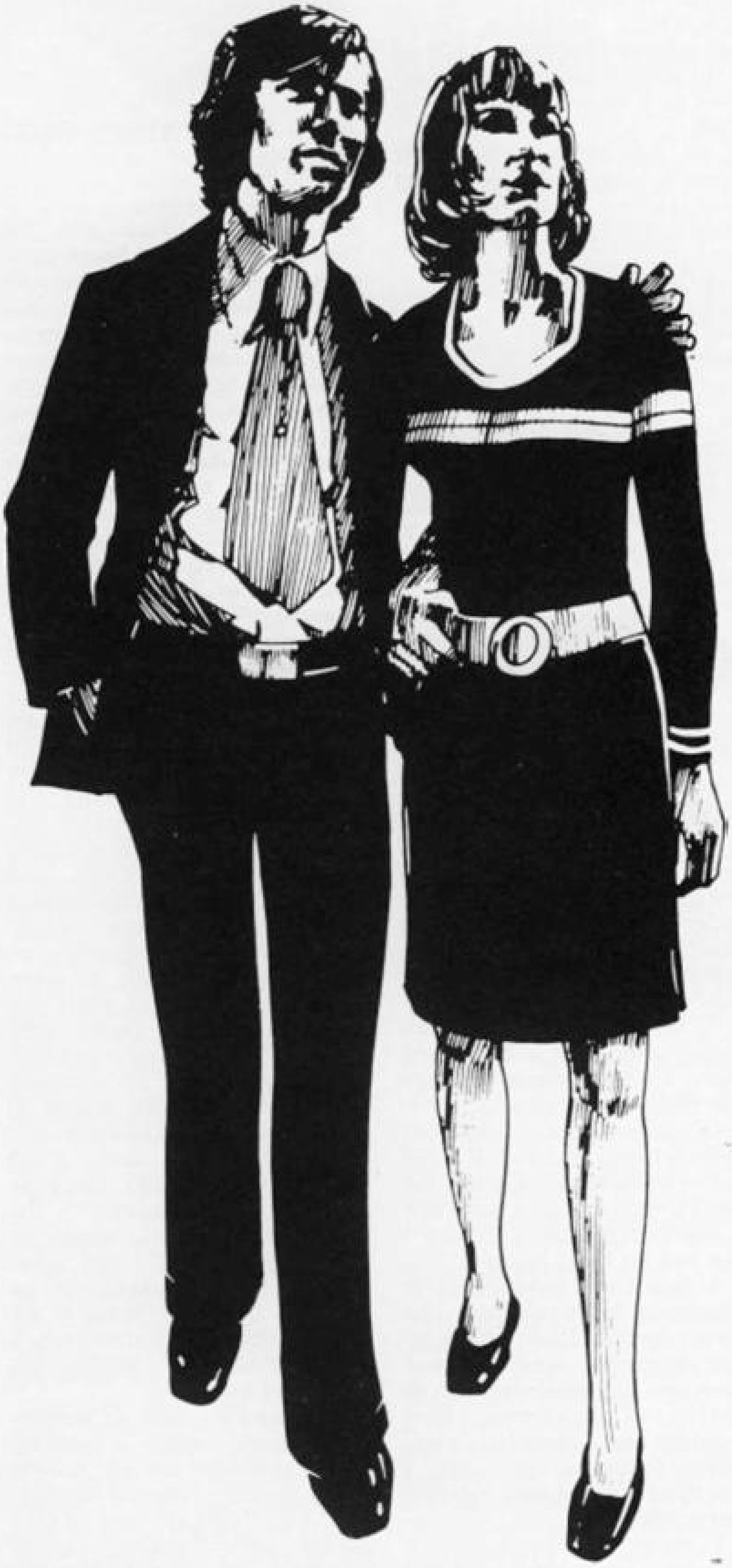
May I make this quite clear to potential buyers of the ZX81 system: when you buy your ZX81, you will in all probability get a 0.7A PSU with your machine. At a later date, if you decide to purchase a ZX Printer, you will get another PSU this time with a current rating of 1.2A. You will therefore have a superfluous PSU which you will have paid for.

I recently spoke to a spokesperson from WH Smiths who made no secret of the fact that consumers would end up with two PSUs and one of them would probably be redundant. It was explained to me that WH Smiths have to accept packages from Sinclair Research 'as is' and therefore have to sell them as such.

This experience brings me to two conclusions: Uncle Clive is severely lacking in consumer relations and WH Smiths should train their staff properly!

I would be most interested to hear other Sinclair users' views on these matters.
Yours faithfully,

Alan Turnbull,
Stockport,
Cheshire.







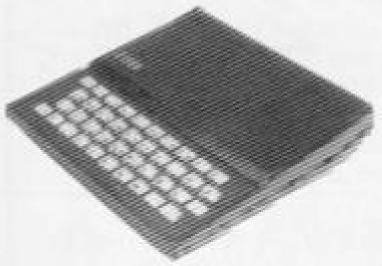
ZX tipster

Dear ZX Computing,
While looking through D Jones' article on tips to improve your programs, I found this:

```
10 POKE 16418,0
20 PRINT AT 23,5;"
   ■■■■■■■■■■"
30 POKE 16384,74
```

Can anyone improve on this?
Yours faithfully,

J Crawford,
Beccles,
Suffolk.



ROM bug?

Dear ZX Computing,
On writing a program for my ZX81, I think I have discovered a ROM bug. Consider the following program.

```
10 FOR N=20 TO 30
20 LET A = SQR N
30 PRINT A;"--5"
40 IF A=5 THEN PRINT "THEY
   ARE EQUAL"
50 NEXT N
```

As you can see, when N attains the value of 25, SQR N should be 5 and the screen should display the message from line 40.

However, this does not happen. I have even tried this on a friend's ZX81 just to be certain it was not a malfunction on my own machine.

The problem can be solved if you write:

```
40 IF INT A=5 THEN PRINT
   "THEY ARE EQUAL"
```

But then the next declarations won't be accurate. The best solution would be to use the following lines:

```
20 LET A$ = STR$ SQR N
30 PRINT A$;"--5"
40 IF A$ = "5" THEN PRINT
   "THEY ARE EQUAL"
```

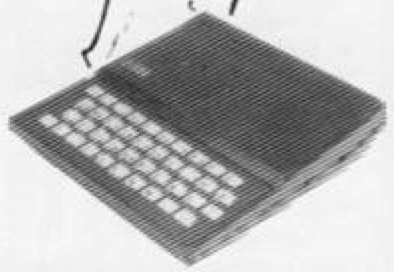
To be certain that the problem is not from the FOR. . . NEXT instruction, try the following with the first program.

```
5 LET N = 19
10 LET N = N+1
50 GOTO 10
```

Hoping to have contributed to the war against bugs.
Yours faithfully,

Paulo Ricardo Plath Xavier,
Bairro Dr Augusto de Castro,
Lote 9 2°B,
2780 Oeiras,
Portugal.





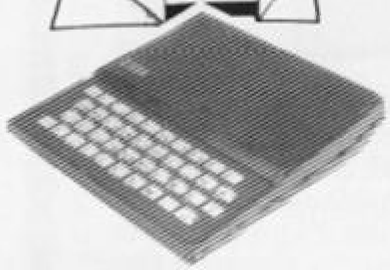
What's your problem?

Dear ZX Computing,
I would like to ask your advice concerning the 16K RAM Pack. I recently borrowed a 16K RAM Pack from a friend but I have a lot of problems trying to load my programs.

I have tried various volume and tone settings, but nothing seems to load. When I load 1K programs, I have to disconnect the RAM Pack. Please help me.
Yours faithfully,

Paul Gingell,
Killiney,
County Dublin,
Ireland.

● *In this issue, Paul, we have included an article covering some of the problems of LOADing and SAVEing which I hope will be of some help to you. Perhaps you may like to try the tip explained in the letter from Oswald Baruch.*



How long?

Dear ZX Computing
My interest having been aroused by your program 'Getting primed' (*ZX Computing*, Aug/Sept 1982, page 73), I decided to try the program.

In Slow mode, after 47 hours and 10 minutes, without any signs of over-heating, the printer came with your 10,000th prime, and the ZX carried on, as happy as ever. I modified the program so that only primes 9990 to 10010 were printed, the others were only displayed on the screen, which I turned off most of the time.

I was once again impressed by what this little machine can do.
Yours faithfully,

Dr. Dick Zeilstra,
Zu den Rehwiesen 9,
4100 Duisburg 1,
W. Germany.

9990	104651
9991	104659
9992	104677
9993	104681
9994	104683
9995	104693
9996	104701
9997	104707
9998	104711
9999	104717
10000	104723
10001	104729
10002	104743
10003	104759
10004	104761
10005	104773
10006	104779
10007	104789
10008	104801
10009	104803
10010	104827

Simple as 1, 2, 3

Dear ZX Computing,
I was about to join all those who are complaining about the difficulties with LOADing. However, in the wake of my 'experiments' to get this operation more reliable, I am persuaded that all is not lost. I have discovered a way of checking the quality of a recording, without losing the program.

Carry out the following:

1) After writing and debugging a program, save it on a cassette, preferably 'from within' (with the SAVE as a program instruction).
2) Do not type NEW.
3) Type LOAD and the program name.
4) Rewind the tape to the beginning of the recording, adjust the volume to the correct level (or what you assume to be the correct level) and start the tape.
5) Press Newline and watch the screen.
6) If the recording on the tape is good, the program will load normally into the computer, and often the loading will start to work if recorded 'from within'.
7) If the recording is faulty, there are two possible things that could happen:
i – during loading, the silence pattern appears on the screen.
ii – at the end of the loading, the silence pattern appears on the screen.
8) If the silence pattern shows up, press the Break key. The program which was previously in the computer will reappear either immediately or after pressing Newline.
9) Now, you can attempt a new SAVE.
10) Repeat the above steps until all check up to point six are positive. Now, you have really SAVEd your program.
Good luck!
Yours faithfully,

Oswald Baruch,
20306 Nesher – Givat Amos,
PO Box 1049,
Israel.

Numerology

Gary Nugent, from Dublin, presents a program which will tell you a little bit more about your friends. All you have to do is type their names in, the ZX81 does the rest.

Numerology is a program written for a ZX81 with 16K RAM Pack which will reveal aspects of a person's character from their name.

Simply type out your name (or anyone else's), separating each name – Christian name and surname – by a space, followed by Newline. Hyphenated names and initials are also catered for. The computer will then calculate the Full Name Number by allotting each letter of the alphabet with a number. Based on the Full Name Number, you will see a display of a character outline for that person.

Try typing your friends' names in — the results may surprise you!



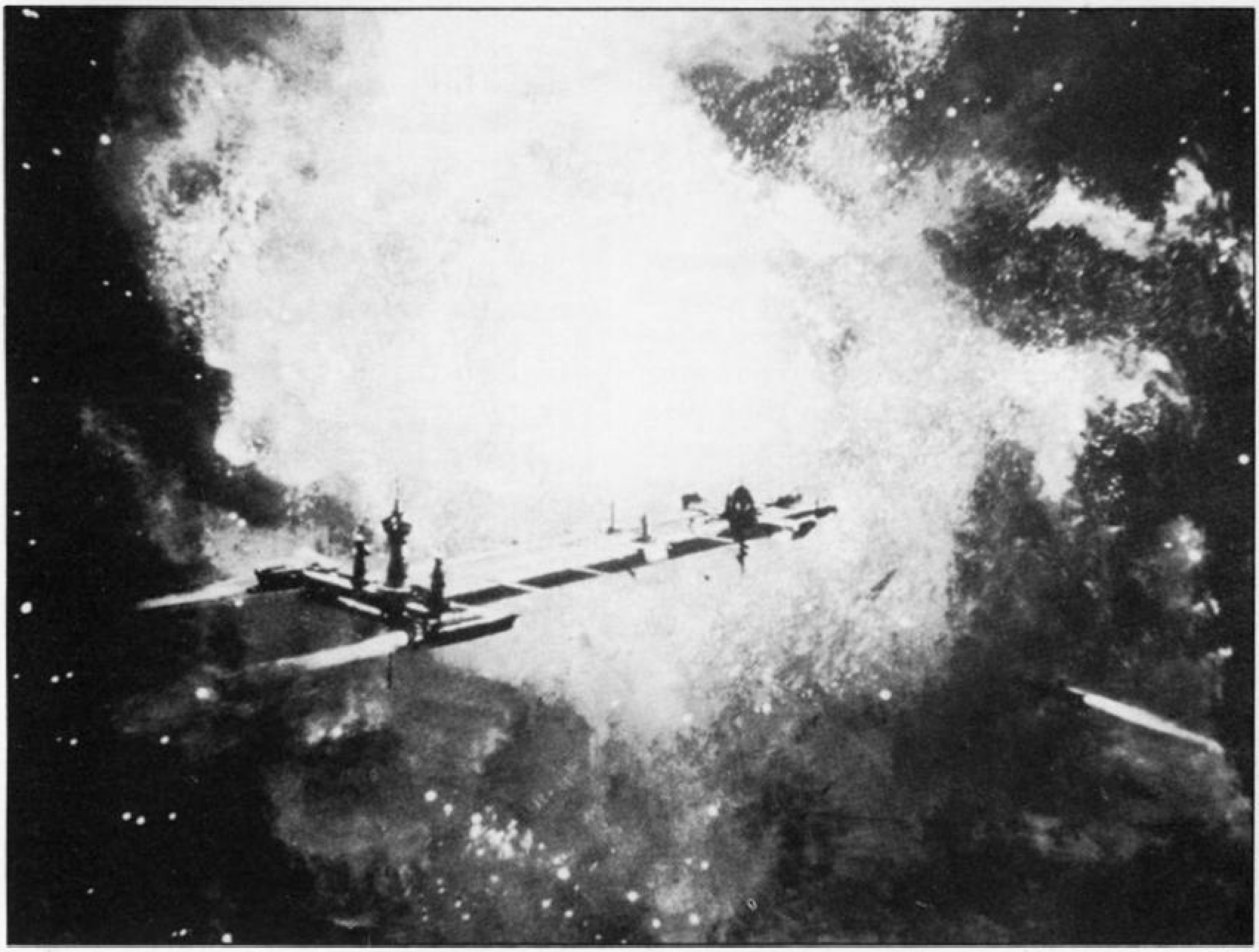

```
   1 REM **  NUMEROLOGY    **
   2 REM **  G. NUGENT     **
   3 REM **                **
  10 REM
  15 CLS
  20 DIM N(26)
  30 FOR I=1 TO 26
  40 LET N(I)=PEEK (16582+I)
  50 NEXT I
  60 PRINT AT 9,3;"PLEASE ENTER
YOUR NAME:"
  70 INPUT N$
  80 LET M$=""
  85 FOR I=1 TO LEN N$
  90 LET M$=M$+CHR$ (CODE N$(I)+
128)
  95 NEXT I
 100 LET X=INT ((32-LEN N$)/2)
 110 FOR I=1 TO 11
 120 PRINT AT 13,X;N$
 122 FOR J=1 TO 6
 123 NEXT J
 125 PRINT AT 13,X;M$
 127 FOR J=1 TO 6
 128 NEXT J
 130 NEXT I
 140 LET NUM=0
 150 FOR I=1 TO LEN N$
 160 IF N$(I)="-" OR N$(I)="." O
R N$(I)=" " THEN GOTO 180
 170 LET NUM=NUM+N(CODE N$(I)-37
)
 180 NEXT I
 190 LET C$=STR$ NUM
 200 LET NUM=0
 210 FOR I=1 TO LEN C$
 220 LET NUM=NUM+VAL C$(I)
 230 NEXT I
 240 IF NUM>9 THEN GOTO 190
 245 CLS
 250 GOSUB NUM*1000
 260 FOR I=1 TO 500
 270 NEXT I
 280 CLS
 290 PRINT AT 10,7;"ANOTHER PERS
ON ?"
 300 LET A$=INKEY$
 310 IF A$="Y" THEN RUN
 320 IF A$<>"N" THEN GOTO 300
 330 STOP
1000 PRINT TAB 7;" NO. 1 PERSON 
"
1010 PRINT AT 3,0;"  YOU HAVE AN
 ALMOST BLIND EX-"
1020 PRINT "HUBERANCE, A SELF-EX
CITEMENT,"
1030 PRINT "AN EFFERVESCENCE, AN
AUDACITY."
1040 PRINT "PLEASE CALM DOWN A B
IT."
1050 PRINT "YOU CAN SURPASS ALL 
OTHERS IN"
1060 PRINT "EASY CHARM. ON THE W
HOLE YOU"
1070 PRINT "ARE POSITIVE, HUMORO
US, THE"
1080 PRINT "POSSESSOR OF ENDURIN
G ENERGY."
1090 PRINT "ON THE OTHER HAND AT
 YOUR WORST"
1100 PRINT "YOU CAN BE SELF-ABSO
RBED TO THE"
1110 PRINT "POINT OF BEING POMPO
US."
1120 PRINT
1130 PRINT " WORK  YOU WORK WELL
 WITH YOUR"
1140 PRINT "HANDS. YOU ALSO DO W
ELL IN"
1150 PRINT "SCIENCES, PHILOSOPHY
, ECONOMICS"
1160 PRINT "AND LAW AND YOU WOUL
D MAKE AN"
1170 PRINT "EXCELLENT TEACHER."
1180 FOR I=1 TO 750
1190 NEXT I
1200 CLS
1210 PRINT TAB 7;" OTHER NO. 1'S
 "
1220 PRINT AT 3,7;"HUMPHREY BOGA
RT";TAB 7;"CHARLIE CHAPLIN";TAB 
7;"ALBERT EINSTEIN";TAB 7;"SOCRA
TES";TAB 7;"JANE FONDA";TAB 7;"J
ACK JONES";TAB 7;"FRANK COUSINS"
;TAB 7;"PRINCE CHARLES"
1230 RETURN
2000 PRINT TAB 7;" NO. 2 PERSON 
"
2010 PRINT AT 3,0;"  YOU ARE CHA
RMING, SUBTLE AND"
2020 PRINT "HUMOROUS, BUT YOU AL
SO HAVE A"
2030 PRINT "CAPACITY FOR CREATIN
G EXCEPT-"
2040 PRINT "IONAL DISCORD. YOUR 
MAJOR FAIL-"
2050 PRINT "ING IS AN INABILITY 
TO PUT PLANS"
2060 PRINT "INTO ACTION. SECOND 
THOUGHTS,"
2070 PRINT "INDECISION, A LACK O
```

```
F ENERGY"
2080 PRINT "CONSPIRE TO ROB VITA
LITY AND"
2090 PRINT "CONFIDANCE. YOU ARE
THE COMPROM-"
2100 PRINT "ISER WITH A DANGEROU
S TENDANCY"
2110 PRINT "TO RETIRE INTO FANTA
SY. BUT IF"
2120 PRINT "DISCIPLINE PUTS HARD
 EDGES ON"
2130 PRINT "CLOUDS YOU CAN DISPL
AY PRODIG-"
2135 PRINT "IOUS ABILITIES."
2140 PRINT
2150 PRINT " WORK: YOU ARE ARTIS
TS, ATHLETES"
2160 PRINT "PERFORMERS OF ALL KI
NDS AND GOOD"
2170 PRINT "SALESMEN."
2180 FOR I=1 TO 750
2190 NEXT I
2200 CLS
2210 PRINT TAB 7;" OTHER NO. 2'S
 "
2220 PRINT AT 3,7;"KARL MARX";TA
B 7;"PABLO PICASSO";TAB 7;"BING
CROSBY";TAB 7;"JEAN-PAUL SARTRE"
;TAB 7;"BENJAMIN BRITTEN";TAB 7;
"GENGHIS KHAN";TAB 7;"DR. CRIPPE
N";TAB 7;"LEN MURRAY";TAB 7;"HAR
OLD WILSON"
2230 RETURN
3000 PRINT TAB 7;" NO. 3 PERSON 
"
3010 PRINT AT 3,0;"  THIS IS THE
 WORLDLY NUMBER."
3020 PRINT "THREES ARE POWERFUL,
 QUICK AND"
3030 PRINT "PROUD, JOVIAL AND EX
PANSIVE AND"
3040 PRINT "ALERT, ON THE SIDE O
F LAW AND"
3050 PRINT "ORDER, ACCUSTOMED TO
 THE EXER-"
3060 PRINT "CISE OF AUTHORITY, C
APABLE"
3070 PRINT "PEOPLE FOR WHOM THIN
GS GO WELL."
3080 PRINT "BUT THEY HAVE OTHER
FAILINGS, AS"
3090 PRINT "WHICH OF US DO NOT?
THREES CAN"
3100 PRINT "BE INSUFFERABLE - IN
SENSITIVE,"
3110 PRINT "HEAVYHANDED AND SUPE
RFICIAL."
3120 PRINT
3130 PRINT " WORK: YOU WILL DO W
ELL IN THE"
3140 PRINT "ARMY, THE GOVERNMENT
 OR A MULTI-"
3150 PRINT "NATIONAL INDUSTRY."
3160 FOR I=1 TO 550
3170 NEXT I
3180 CLS
3190 PRINT TAB 7;" OTHER NO. 3'S
 "
3200 PRINT AT 3,7;"T. E. LAWEREN
CE";TAB 7;"BARBARA CASTLE";TAB 7
;"ARISTOTLE";TAB 7;"GOUGH WHITLA
M";TAB 7;"RONALD REAGAN";TAB 7;"
MICHAEL FOOT"
3210 RETURN
4000 PRINT TAB 7;" NO. 4 PERSON 
"
4010 PRINT AT 3,0;"  FROM A WORL
DLY STANDPOINT,"
4020 PRINT "FOURS ARE PROBABLY T
HE LEAST"
4030 PRINT "ABLE TO SUCCEED - SO
RRY ABOUT"
4040 PRINT "THAT. ON THE OTHER H
AND YOU"
4050 PRINT "DON""T SEEM TO CARE
MUCH. YOU"
4060 PRINT "ARE SENSITIVE, IDEAL
ISTIC,"
4070 PRINT "ORIGINAL AND WARM, P
ERCEPTIVE"
4080 PRINT "TO A FAULT, ALWAYS S
EEING BOTH"
4090 PRINT "SIDES OF EVERY QUEST
ION. YOU ARE"
4100 PRINT "LOYAL AND NICE, BUT
ALSO OFTEN"
4110 PRINT "CONTRARY AND TOUCHY.
"
4120 PRINT
4130 PRINT " WORK: YOUR APTITUDE
 IS FOR THE"
4140 PRINT "ARTS AND, STRANGELY
ENOUGH, FOR"
4150 PRINT "POLITICS."
4160 FOR I=1 TO 550
4170 NEXT I
4180 CLS
4190 PRINT TAB 7;" OTHER NO. 4'S
 "
4200 PRINT AT 3,7;"ORSON WELLES"
;TAB 7;"NORMAN MAILER";TAB 7;"SP
IKE MILLIGAN";TAB 7;"NAPOLEAN BO
NAPARTE";TAB 7;"STANLEY KUBRICK"
;TAB 7;"JIM CALLAGHAN";TAB 7;"PR
ESIDENT FORD";TAB 7;"SOOTY"
4210 RETURN
5000 PRINT TAB 7;" NO. 5 PERSON 
"
5010 PRINT AT 3,0;"  THE NUMBER
OF INTELLECT, WIT"
5020 PRINT "AND WORDS, SUPREMELY
 ADAPTABLE"
5030 PRINT "QUICK AND CYNICAL, F
IVE TAKES"
5040 PRINT "THE WORLD AS ITS OYS
TER. ALMOST"
5050 PRINT "EVERYONE LIKES A FIV
E. NOT A"
5060 PRINT "POWERFUL NUMBER, BUT
 A LUCKY"
5070 PRINT "ONE, THE NUMBER OF G
AMES AND"
5080 PRINT "GAMBLING, MENTALLY A
GILE, UN-"
5090 PRINT "SENTIMANTAL, VOLATIL
E, HIGHLY"
5100 PRINT "STRUNG. YOU HAVE A Q
UICK TEMPER"
5110 PRINT "AND YOUR CRAVING FOR
 EXCITEMENT"
5120 PRINT "AND CHANGE CAN LEAD
TO PARANOIA"
5130 PRINT "AND EXHAUSTION."
5140 PRINT
5150 PRINT " WORK: THE BEST OF S
AINTS, THE"
5160 PRINT "BEST OF CROOKS, GOOD
 SPIES,"
5170 PRINT "SALESMEN, GAMBLERS A
ND WRITERS."
5180 FOR I=1 TO 550
5190 NEXT I
5200 CLS
5210 PRINT TAB 7;" OTHER NO. 5'S
 "
5220 PRINT AT 3,7;"BOBBY FISHER"
;TAB 7;"VINCENT VAN GOGH";TAB 7;
"ROBIN DAY";TAB 7;"WILLIAM SHAKE
SPEARE";TAB 7;"JAMES BOND";TAB 7
;"ELTON JOHN";TAB 7;"FIDEL CASTR
O";TAB 7;"CHE GUEVARA";TAB 7;"ST
. PAUL OF TARSUS"
5230 RETURN
6000 PRINT TAB 7;" NO. 6 PERSON 
"
6010 PRINT AT 3,0;"  PRACTICAL,
SINGLE-MINDED, IN-"
6020 PRINT "DUSTRIOUS, MAGNETIC,
 THE BORN"
6030 PRINT "LEADER, SIXES ATTRAC
T WARM"
6040 PRINT "DEVOTION. SIXES ARE
PRAGMATIC."
6050 PRINT "AND TIRELESSLY RESPO
```

```
NSIBLE. THEY"
6060 PRINT "GET THINGS DONE. BUT THEIR VIEW"
6070 PRINT "OF LIFE IS OFTEN BLINKERED. THEY"
6080 PRINT "CAN BE RUTHLESS AND INSENSITIVE,"
6090 PRINT "SMOOTHLY PUSHING OPPOSITION FROM"
6100 PRINT "THEIR PATH."
6110 PRINT
6120 PRINT " WORK: THEY DO WELL IN ALMOST"
6130 PRINT "ANYTHING. LAW IS PARTICULARLY"
6140 PRINT "GOOD. SO ARE MOST FIELDS OF"
6150 PRINT "BUSINESS."
6160 FOR I=1 TO 550
6170 NEXT I
6180 CLS
6190 PRINT TAB 7;" OTHER NO. 6""S █"
6200 PRINT AT 3,7;"MARGARET THATCHER";TAB 7;"EDWARD HEATH";TAB 7;"INDIRA GHANDI";TAB 7;"DANNY LA RUE";TAB 7;"BRIGGITE BARDOT";TAB 7;"JIM SLATER";TAB 7;"ALBERT FINNEY";TAB 7;"MAO TSE TUNG"
6210 RETURN
7000 PRINT TAB 7;" NO. 7 PERSON "
7010 PRINT AT 3,0;"█ SEVENS CAN BE ACTIVE AND THEY"
7020 PRINT "CAN BE SOLITARY, THEY CAN BE"
7030 PRINT "ARTISTIC AND THEY CAN BE INART-"
7040 PRINT "ICULATE. THEY ARE USUALLY GOOD"
7050 PRINT "LOOKING AND ATTRACTIVE, BUT THEY"
7060 PRINT "ARE OFTEN SHY AND SEXUALLY IN-"
7070 PRINT "HIBITED. STILL, YOU SEVENS GET"
7080 PRINT "THINGS DONE, ARE VERY RELIABLE"
7090 PRINT "AND DON""T MESS AROUND."
7100 PRINT
7110 PRINT " WORK: YOU WOULD ENJOY A LIFE"
7120 PRINT "ABROAD WITH TRAVEL. GO TO SEA,"
7130 PRINT "YOU""LL DO WELL THERE, ON THE"
7140 PRINT "OTHER HAND YOU COULD TURN OUT TO"
7150 PRINT "BE A MUSICIAN, A POET OR A"
7160 PRINT "PAINTER, A MYSTIC OR A CLERGY-"
7170 PRINT "MAN. IF YOU GO INTO BUSINESS"
7180 PRINT "YOU WILL PROBABLY END UP BOSS."
7190 FOR I=1 TO 600
7200 NEXT I
7210 CLS
7220 PRINT TAB 7;" OTHER NO. 7""S █"
7230 PRINT AT 3,7;"HENRY KISSINGER";TAB 7;"GERMAINE GREER";TAB 7;"ABRAHAM LINCOLN";TAB 7;"MICK JAGGER";TAB 7;"GEORGE BERNARD SHAW";TAB 7;"MALCOLM MUGGERIDGE";TAB 7;"TONY BENN"
7240 RETURN
8000 PRINT TAB 7;" NO. 8 PERSON "
8010 PRINT AT 2,0;"█ A NUMBER OF INCREDIBLE, INEX-"
8020 PRINT "ORABLE POWER, EIGHT PONDERS"
8030 PRINT "SLOWLY, MOVE IMPLACABLY, AND"
8040 PRINT "STARES INTO THE BLACKEST ABYSS"
8050 PRINT "OF THEM ALL. THEY ARE AMBITIOUS,"
8060 PRINT "INTIMIDATING, INTENSE, SELF-"
8070 PRINT "RIGHTOUS WITH A PASSION FOR"
8080 PRINT "JUSTICE. THEY TEND TO GREAT"
8090 PRINT "SUCCESS AND FAILURE, ARE OFTEN"
8100 PRINT "ALONE AND MISUNDERSTOOD. TO"
8110 PRINT "SOME EYES EIGHT IS THE LEAST"
8120 PRINT "ATTRACTIVE NUMBER BUT APPARENTLY"
8130 PRINT "THERE ARE GREAT COMPENSATIONS."
8140 PRINT
8150 PRINT " WORK: THEY ARE THE SCHOLARS"
8160 PRINT "HISTORIANS AND PHILOSOPHERS,"
8170 PRINT "THOUGH THEY ALSO DO WELL IN"
8180 PRINT "COALMINES AND ON FARMS. THEY"
8190 PRINT "SO ARE HAPPY IN PUBLISHING, RE-"
8200 PRINT "VIEWING AND INSPECTING."
8210 FOR I=1 TO 750
8220 NEXT I
8230 CLS
8240 PRINT TAB 7;" OTHER NO. 8""S █"
8250 PRINT AT 3,7;"WINSTON CHURCHILL";TAB 7;"ERIC MORECAMBE";TAB 7;"RICHARD BURTON";TAB 7;"J. B. PRIESTLEY";TAB 7;"JOE BUGNER";TAB 7;"LEONID BREZHNEV";TAB 7;"IDI AMIN"
8260 RETURN
9000 PRINT TAB 7;" NO. 9 PERSON "
9010 PRINT AT 3,0;"█ DYNAMIC, STRONG, MAGNETIC, YOU"
9020 PRINT "STRIDE THROUGH LIFE AND OTHERS"
9030 PRINT "STAND ASIDE. YOU ARE LARGER THAN"
9040 PRINT "LIFE, LUCKY, CAPABLE, EFFECTIVE"
9050 PRINT "AND ALWAYS IN THE THICK OF IT."
9060 PRINT "YOU ARE IMPULSIVE AND QUICK-"
9070 PRINT "TEMPERED TOO. STILL THAT CHAR-"
9080 PRINT "ISMA CHARMS ALMOST EVERYONE. YOU"
9090 PRINT "CAN HAVE BLACK MOODS, BUT WHEN"
9100 PRINT "YOU ARE FEELING CONFIDENT (WHICH"
9110 PRINT "IS MOST OF THE TIME) NOTHING CAN"
9120 PRINT "STAND IN YOUR WAY."
9130 PRINT
9140 PRINT " WORK: YOU MAKE FINE SOLDIERS,"
9150 PRINT "ATHLETES AND POLITICIANS,"
9160 FOR I=1 TO 600
9170 NEXT I
9180 CLS
9190 PRINT TAB 7;" OTHER NO. 9""S █"
9200 PRINT AT 3,7;"JACK KENNEDY";TAB 7;"ELVIS PRESLEY";TAB 7;"EZRA POUND";TAB 7;"TOM JONES";TAB 7;"RUDOLPH VALENTINO";TAB 7;"RALPH NADER";TAB 7;"JOHN BETIEMAN";TAB 7;"JACKIE ONASSIS";TAB 7;"THE QUEEN"
9210 RETURN
```

Space adventure

Fourteen year old David Lambert delves into outer space with a great program for the 16K ZX81. All you have to do is reach Earth before the aliens get you . . .



Photograph courtesy of Walt Disney Productions

Space Adventure is a game for the ZX81 with 16K RAM Pack. You are travelling in a space ship journeying back to mother Earth. The trouble is that there are a number of nasty aliens who are out to get you!

You first choose how long you wish the game to last, then your fuel and estimated time of arrival are calculated in lines 3 and 4. When an alien approaches, you will get a warning on screen and it is then up to you to choose your course of action. You can either increase your speed, take evasive action, enter hyperspace or, as a last resort, scuttle your ship. However, if you take evasive action, you will use up more fuel than if you increased your speed. You only have a limited amount of fuel so care must be taken in your choice. If you have not enough fuel to finish the run to Earth, you will get a message on the screen warning you of this and you will have to leap into hyperspace to get your estimated time of arrival down.

Your star ship also has the capability to fire missiles at any annoying aliens, but be warned, if you score a direct hit the aliens will not always be blown up — they might have their protective shields up to deflect your missile. If they do have their shields up, it is very likely they will fire at you so be prepared for quick evasive action.

All's well . . .

There are several ways for the game to end: your ship could be blown up; you could scuttle your own ship; you could run out of fuel; your engines could overheat; or the explosive missiles you are carrying could accidentally be triggered. Of course, if you manage to reach Earth after your journey you will be rewarded with a 'Well done' message which is generated in lines 2070 to 2130. You will also be told how many aliens you managed to do away with on your trip home.

Here is a table of the variables used in the program to help you follow how David has put the listing together.

F —	Fuel.
E —	Estimated time of arrival.
K —	Kills.
T —	Temperature.
TRAV —	Game length.



```
   1 PRINT "HOW LONG TO TRAVEL
 1 TO 30"
   2 INPUT TRAV
   3 LET F=TRAV*10
   4 LET E=TRAV*9
   5 LET K=0
   6 LET T=30
   7 CLS
  10 PRINT "YOU ARE THE COMANDER
 OF STAR "
  20 PRINT "SHIP  (R1 DICULT). I
T IS ON ITS  WAY TO PLANET TAERH
. WE KNOW    THIS PLANET AS EART
H. YOU ARE   SHIPING EXPLOSIVES,
. THEY       COULD GO OFF AT ANY
 MINUTE"
  30 PRINT " YOU COULD MEET UP W
ITH LONGTEMS ON THE WAY. THEY AR
E NOT VERY   NICE ALIENS TO MEET
 UP WITH IF  THEY GET THE CHANCE
 FIRST THEY  WILL OPEN FIRE ON Y
OU. YOU WILL HAVE PRIRE WORNING
BEFORE THIS  HAPPENS. "
  31 PRINT "BEWARE. WHEN YOU TAK
E EVASIVE    ACTION YOU USE UP.L
OTS OF FUEL  THE ONLY WAY TO GET
 THIS BACK   IS TO GO INTO HYPER
SPACE DRIVE  THIS WILL ADD ON TO
 YOUR ETA."
  40 PRINT "          PRESS ANY KEY T
O START."
  50 IF INKEY$="" THEN GOTO 50
  60 CLS
  70 PRINT ".        :     *    .
"
  80 PRINT "
 .    *
"
  90 PRINT "                     ;
"
 100 PRINT "   O             .
"
 110 PRINT "
 120 PRINT "   ..
 122 PRINT AT 15,0;"  COMPUTER R
EAD OUT"
 123 PRINT AT 13,0;"PRESS T IF T
EMP IS 50 OR OVER"
 130 PRINT AT 17,0;"FUEL  ETA  K
ILLS  TEMP"
 140 PRINT AT 18,0;F;AT 18,6;E;A
T 18,11;K;AT 18,18;T
 145 LET F=F-4
 147 LET E=E-4
 150 LET RND=INT (RND*6)
 151 IF E>F THEN PRINT AT 16,0;"
NOT ENOUGH FUEL TO REACH ETA."
 152 LET EX=INT (RND*20)
 153 IF EX=5 THEN GOTO 651
 154 IF T>=70 THEN GOSUB 3300
 155 IF F<=0 THBN GOTO 1500
 156 IF E<=0 THEN GOTO 2000
 160 IF RND=4 THEN GOTO 180
 170 GOTO 130
 180 FOR A=1 TO 10
 190 PRINT AT 20,0;"  ALIEN IN S
CANNER RANGE"
 200 PRINT AT 20,0;"  ALIEN IN S
CANNER RANGE"
 210 NEXT A
 215 CLS
 220 FOR A=1 TO 5
 221 CLS
 222 PRINT "     .     *     :
   :  "
 223 PRINT "     .     ;     :
   *  "
 224 PRINT
 225 PRINT "     .    ;     *
  O   "
 226 PRINT "     .          :
    O  "
 230 PRINT AT 10,0;"  *     .
"
 240 PRINT AT 11,0;"     O
    .   *  "
 250 PRINT AT 12,0;" ..     *
       +  "
 260 PRINT AT 14,0;"
"
 270 NEXT A
 280 PRINT
 290 PRINT "        RED  ALERT"
 300 PRINT "    ALIEN. YOUR OPTIO
NS ARE"
 310 PRINT "(1) TAKE EVASIVE ACT
ION."
 320 PRINT "(2) INCREASE SPEED."
 330 PRINT "(3) SCUTTLE"
 340 PRINT "(4) HYPERSPACE DRIVE
"
 350 IF INKEY$="1" THEN GOTO 500
 360 IF INKEY$="3" THEN GOTO 600
 370 IF INKEY$="2" THEN GOTO 380
 371 IF INKEY$="4" THEN GOTO 300
0
 375 GOTO 350
 380 CLS
 381 PRINT "INPUT NEW SPEED"
 390 INPUT SP
 400 LET RET=INT (RND*3)
 410 IF RET=2 THEN CLS
 420 IF RET=2 THEN PRINT "ALIEN
CLOSEING"
 430 IF RET=2 THEN GOTO 350
 440 LET RET=INT (RND*9)
 450 IF RET=5 THEN CLS
 460 IF RET=5 THEN PRINT "YOUR R
EACTER HAS BLOWN UP"
 465 IF RET=5 THEN GOTO 650
 470 CLS
 480 GOTO 70
 500 CLS
 505 PRINT "YOU ARE TAKING EVASI
VE ACTION"
 510 PRINT "YOU OPTIONS ARE."
 520 PRINT "(1) FIRE MISSILE"
 530 PRINT "(2) INCREASE SPEED"
 540 PRINT "(3) FIRE LAZERS"
```

```
 550 IF INKEY$="1" THEN GOTO 700
 560 IF INKEY$="2" THEN GOTO 800
 570 IF INKEY$="3" THEN GOTO 900
 590 GOTO 550
 600 CLS
 610 PRINT "YOU ARE GOING TO KIL
L ALL THE"
 620 PRINT "MEMBERS OF YOUR CREW
ARE YOU "
 630 PRINT "SURE YOU WANT TO DO
THAT"
 640 INPUT S$
 650 IF INKEY$="N" THEN GOTO 70
 651 CLS
 652 IF EX=5 THEN PRINT "YOUR EX
PLOSIVES WENT OFF........"
 653 PAUSE 100
 660 CLS
 663 FOR A=1 TO 6
 664 CLS
 665 PRINT AT 10,10
 666 PRINT "          ((((()))))"
 670 PRINT "
 680 PRINT "
 685 PRINT "
 686 PRINT "          ((((()))))"
 690 NEXT A
 692 PRINT "YOU ARE DEAD........
....."
 694 STOP
 700 CLS
 710 PRINT AT 17,0;"
 720 PRINT AT 18,0;"
 730 PRINT AT 19,0;"
"
 740 PRINT AT 20,0;"
 750 FOR A=1 TO 21
 760 SCROLL
 770 NEXT A
 777 LET F=F-5
 780 LET RND=INT (RND*10)
 790 GOTO 810
 800 GOTO 350
 810 IF RND=5 THEN PRINT "ALIEN
HAD SHIELD UP"
 811 IF RND=5 THEN SCROLL
 820 IF RND=5 THEN PRINT "HE NOW
FIRES AT YOU."
 830 IF RND=5 THEN GOTO 1000
 831 CLS
 832 FOR C=1 TO 6
 833 PRINT AT 8,15;"
 834 PRINT AT 9,15;"
 835 PRINT AT 10,15;"
 836 CLS
 837 NEXT C
 840 PRINT
 845 PRINT
 850 PRINT "ALIEN NOW GONE."
 860 PRINT "YOU CAN KNOW CONTINU
E WITH TRIP"
 862 PRINT "BUT REMEMBER THAT YO
U HAVE USED"
 863 PRINT "ALOT OF FUEL. YOUR T
EMP HAS GONE UP ASWELL."
 864 LET F=F-10
 865 LET K=K+1
 866 LET T=T+5
 867 IF INKEY$="" THEN GOTO 867
 870 GOTO 60
 900 CLS
 910 FOR J=0 TO 30
 920 PRINT AT 10,J;"■"
 930 PRINT AT 11,J;"■"
 940 NEXT J
 950 GOTO 780
1000 FOR A=1 TO 100
1010 NEXT A
1020 CLS
1030 PRINT AT 17,0;"
1040 PRINT AT 18,0;"
1050 PRINT AT 19,0;"
"
1060 PRINT AT 20,0;"
1070 FOR A=1 TO 21
1080 SCROLL
1090 NEXT A
1091 LET F=F-5
1100 GOTO 650
1500 CLS
1510 PRINT "YOU HAVE RUN RIGHT O
UT OF FUEL"
1520 PRINT "YOU WILL DRIFT IN SP
ACE FOR"
1530 PRINT "EVER AND EVER.......
..."
1540 PRINT "DO YOU WANT TO TRY A
GAIN"
1550 IF INKEY$="Y" THEN RUN
1560 IF INKEY$="N" THEN STOP
1570 GOTO 1550
2000 CLS
2010 PRINT "YOU HAVE MADE IT TO
TAERH"
2020 PRINT "SCREEN WILL GO BLANK
 WHILE I"
2030 PRINT "THINK. PLEASE DO NOT
 PRESS "
2040 PRINT "BREAK WHILE I AM DOI
NG THIS. "
2050 PRINT "THANKYOU. NOW PRESS
ANY KEY"
2055 IF INKEY$="" THEN GOTO 2055
2060 CLS
2070 FAST
2080 FOR I=-4 TO 4
2090 FOR J=0 TO 120
2100 PLOT 32+20*SIN (I+(J/60*PI)
),22+20*COS (J/60*PI+I/4*PI)
2110 NEXT J
2120 NEXT I
2130 PRINT "WELL DONE"
2140 IF K>0 THEN PRINT "YOU KILL
ED ";K;" ALIENS"
2150 STOP
2700 PRINT "YOUR ENGINES HAVE OV
ER HEATED"
2710 STOP
3000 FOR A=1 TO 6
3001 CLS
3002 PRINT ". ; "
3003 PRINT "  .    .  ;. :   ** "
3004 PRINT " **   O  .    .    ;
:  . "
3010 PRINT "  .     *        :
 *   "
3020 PRINT "   .     .   ;    O
 ; "
3030 PRINT "  .        .    ;
: "
3040 PRINT
3050 PRINT "   .       .     O
:  : "
3060 PRINT
3070 PRINT "   .        **      ;
 :  : "
3080 PRINT " .     .      ;    ::
 :  ."
3090 PRINT
3100 PRINT ". .           +
:  : "
3101 NEXT A
3110 PRINT "DECODING:"
3120 PRINT COS (PI+1+9)
3230 LET E=E-20
3240 PRINT "RETURNING TO COURSE"
3250 PAUSE 50
3255 CLS
3260 GOTO 70
3300 CLS
3310 PRINT "YOU TOOK TO MUCH EVA
SIVE ACTION AND YOU ENGINES HAVE
 OVER        HEATED"
3320 PRINT "THE ENGINES HAVE NOW
 BEEN COOLED"
3330 PAUSE 500
3340 RETURN
```

File card

Keep all your information on file using this excellent program from Jim Enness of Poole.



This file card program was designed to suit almost any requirements of storing useful information using a 16K ZX81. Great effort had been made to keep the programming minimal (to save memory storage space) and keep all the required functions. The program had also to be easily usable by anyone, calling for clear instructions and fool-proof entries.

The program allows up to 10K of usable storage space. You can, for example, have 100 file cards, allowing up to 100 characters per card (100 x 100 = 10,000). You may require 50 cards with 200 characters per card, or 250 cards with 40 characters. You may use any combination making up the 10K. You can also choose less than this, eg 50 cards at 150 characters if you wish; this shortens the taping out time, or taping back time, which can take up to seven minutes with a full system.

Running the program is quite straightforward. Firstly, you will be asked to give the program a name. This is used for the taping out routine and the Report Page. Still in the Setting Up mode, you will next be asked how many cards you require and your limit of maximum characters per card.

This is followed by a Section Entries mode. Each file card is broken up into sections, eg Name, Address, County, Telephone Number, etc. You will therefore be asked how many sections are required. In the above example, we would need four of course, but the cards may equally be used for Stock Number, Part Description, Price, Quantity or Song Title, Record Label, Artist, etc. The number of sections is only limited to the number of characters allowed per card, and extra sections can, of course, be added in later on in the program.

What's in a name?

The program will then ask you to name each section one at a time. The section names as in the first example, Name, Address, etc, should be kept down to 15 characters in length. This is because these names are used as keywords for the Card Entries mode and longer names will simply get their ends chopped off after 15 characters. No harm can be done by this and you can change the section names later on should you enter a longer name. Punctuation, inverse characters and graphics may be used in these names or on the Card Entries if required.

Next comes the Card Entries mode. It is a good idea to only enter one or two cards first time around to check the system, and enter the bulk of the information later. After answering the number of cards required, the computer will ask you to input information on each section name in turn, onto Card 1. This continues until the number of cards required are completed.

Entries onto the cards is fairly straightforward – you simply enter the words or characters into the string prompt, then press Newline. This is the case with all inputs. The length of each entry is only limited to 255 characters (or the number of characters allowable on the card, should this be less). For example, if we have a section called Name and we enter 'Joe Bloggs' then this section will build up one character at a time until it is ten characters wide to accept this entry including the space. A space 10 characters wide will now be assumed on every card for the 'Name' section.

If on another card, a longer name should be entered, say 'David Appleby' then this section will expand until it reaches the end of the name. It will pad out the 'Joe Bloggs' and shorter names entered with blank spaces behind, shifting any following sections along the card so that they all line up. Each section, therefore, will assume the greatest length entry into that section for its number of characters width. It is because of this system that when entering the first long entry into a section, say an address 30 characters long, a short time (a second or two) will be taken to expand the section length. After this, entries to following cards will be fairly quick, as a shorter address will go straight in and a longer address will only expand the section by a few characters.

You could also use the system as a notebook if you want to, having 30 or so cards with 255 characters and just one section. Each card may then be used as a page containing up to eight lines (including the prompt quotes) of text.

After Card Entries, a report page tells you: Name of Program, Number of Cards, Maximum Characters (from Setting Up mode), Number of Cards/Sections used and the assumed maximum length of each section to date. Newline sends the program to the Option mode.

On the cards

There are ten main options numbered 0 – 9 which may be used. Card Entries, Card Alteration, Card Delete, Section Entries and Section Name Change, do just as their titles suggest and instructions are printed up by the program for the various modes as they are used (as in all the options). Card Delete, however, will shift all the later cards forward one card thus the numbering will change leaving the spare card at the end of the file. Card List will ask for a card number and display the contents of any single card entered. List All Cards will list the name of all the sections (numbered) then list the first line's worth of characters on each card. A blank space is inserted between each section should there be several sections in the line (as may be the case in Part Numbers, etc).

Card Search will search for, and display all the cards which contain your search word/characters within a particular section. This will display the number of and the first line's worth of characters of the card. You can therefore search for a person's name to get the card number of that entry, or for a county to get names of all entries living in a particular area, etc. The search ends with a 'Finish' just in case your search is fruitless.

Section Trim will cut one character off the section specified. If you decide to abbreviate long entries to leave more room on the card, the section must be trimmed down as it has already assumed a max-

imum length that is no longer required.

Section Order Change will take the section number specified and put it as the first item on your card; the section numbering will be altered accordingly. You can therefore move the sections into any order you wish one at a time. It should be noted that in the Card Listing modes, should the information fill the screen, then Newline will clear the screen and continue the listings until the task is completed.

Slow and Fast modes are available from the Options mode, Fast is useful for listings, Slow for entries. 'T' tapes out the program, puts the recorder to record and play first, then enter 'T'. If the program tapes out before you are prepared, leave it to tape out as it will return to the Option mode after this so you can tape out as many times as you wish.

After all the options, the program will return to the Option mode. If you enter Newline alone then the program will give you back the Report Page mode. The program will not come out of this loop until it is turned off. If you wish to start a new file card program then you must press Break whilst it is busy, then RUN.

Illegal entry

Various error messages and traps are used in the program to prevent illegal entries. If a mistake is made in the programming and an error report comes up, check and alter the program as required then enter a direct command GOTO 130. This will save any information being lost.

An alphanumeric sort of program sorts the contents of the cards from the first section, renumbering each card into alphanumeric order. This part of the program has been put in as an optional extra, although I will advise it is worth the extra work. This option should be entered after the main program has been checked and tested. Break whilst running and add in the extra program lines, then RUN. The program will stop at line 81. Lines 8 to 81 inclusive *must* now be deleted to give you back the 10K of memory for the cards. Then GOTO 1.

Sorting cards tends to be a lengthy business so it will automatically presume the Fast mode when 'S' is selected. Sorting 200 entries, 50 characters long, takes just over three minutes. Obviously, less entries take proportionally less time to sort. The search routine, by the way, in the Fast mode for 200 entries takes about 12 seconds to search and display.

The program

Care should be taken entering this program as the memory saving techniques used make it very complex to read.

The variables I=1, Q=0, J=710, K=800, T=1000, L=2, M=7 and N=20, remain constant throughout the program and simply substitute these numbers to save memory. A$ is dimensioned 30 by 15 characters in length. This holds all the phrases that make up the instructions (the spacings should be followed to get the correct displays). A$(1) and A$(10) are left blank, A$(1) to blank the displays and A$(10) to hold the section names; the latter is updated during the program. Incidentally, this is why section names should be limited to 15 characters.

The main printing routine is the input routine at line 1000. This draws the box, top centre of screen and prints in it; this is then followed by the instructions. If extra instructions are required, the routine may start at lines 900 or 950. Most of the options use this routine and a coded message tells the routine what to print and what the input side of it can accept. H$ holds the coded message and G$ is used as the input string. For example, lines 104 to 108 ask how many cards are required, show the Setting Up mode in the box and expect a numerical answer. If the first character in H$ does not equal '0' then a number is expected. The value of this first character is the maximum acceptable; in this case 'T' = 1000 maximum cards are allowed. The second character in H$ holds the error message code for the print routine. The third character is the mode printed in the box, leaving four phrase codes to make up the instructions, ie 1 – 0 – 3 – 4 = "PLEASE ENTER" "BLANK" (twice) then "THE NUMBER OF" "CARDS REQUIRED". The value of these numbers (+ 1) are taken from A$.

Line by line

Line 102 F$ holds the program name.
Line 108 A is the number of cards.
Line 120 B is the maximum number of characters per card.
Line 124 B$ is dimensioned and holds the cards contents.
Line 92 C$ initially holds the graphic characters of code 1.
C$ is used to hold the length of each section entry made on the card in the form of character codes, so the maximum allowable entry into any one section is restricted to 255 characters.
D$ holds the Section Names.
E$ holds the length of the Section Names (and also uses character codes).
Line 132 Prints the report page.
Line 146 GOSUB K (starts at line 800) is a more simple input G$ routine. The variable P is used as a counter in this routine. It is only used when any listings are required and is reset in the Option mode, line 164. When 22 lines have been completed GOSUB K (800) will CLS after input and add 22 to P allowing the next 22 line page to be displayed.
Lines 150 – 180 Print out the options and are the centre control to which all the other routines return. If a number 0 to 9 is entered then it will send the program to this number * 50 + 200. So the various options programs are numbered in steps of 50 from 200 to 650 in the same order as printed. The exceptions are the Slow/Fast, tape out, sort routine, and the Newline command for the Report Page, which can be seen from lines 170 to 180.
Lines 700 – 708 Reset E,F,G and H. The E and F variables are used to hold the number of the beginning and end characters of a particular section name.
Lines 710 – 714 These are repeatedly used to add up the character codes in E$ until the correct section name is put into A$(10). The G and H variables are used to interrogate the codes of C$ in the same manner to find the beginning and end position of an entry.
V, W, X, Y and Z are used as temporary variables in FOR . . . TO loops, etc, in the various routines ending in a 'don't care' state when the program returns to the Option mode.

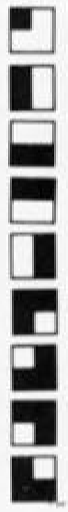
Other variables used are:

C – Number of sections entered.
D – Number of cards used.
R – Number of unused characters left on the card.

Quite a lot of usage is made of the open ended 'TO' as in line 378. I must admit this looks strange but it is quite legal as explained in the Sinclair manual, page 137.

Where you see '*(space)*' leave the number of spaces indicated within the brackets. When you type in some of the other graphics characters within brackets, here is a guide to what you should see on the screen.

Shifted graphic 1 . . . .
Shifted graphic 5 . . . .
Shifted graphic 6 . . . .
Shifted graphic 7 . . . .
Shifted graphic 8 . . . .
Shifted graphic E . . . .
Shifted graphic Q . . . .
Shifted graphic R . . . .
Shifted graphic W . . . .



```
 8 LET I=VAL "1"
1Ø LET Q=I-I
12 LET J=VAL "71Ø"
14 LET K=VAL "8ØØ"
16 LET T=VAL "1ØØØ"
18 LET L=I+I
2Ø LET M=VAL "7"
22 LET N=VAL "2Ø"
24 DIM A$(M+N/L,N-M+L)
26 LET A$(L)="PLEASE ENTER -"
28 LET A$(L+I)="(3 spaces) THE NAME OF"
3Ø LET A$(L+L)="(1 space) THE NUMBER OF"
32 LET A$(N-L)="CARDS REQUIRED."
34 LET A$(M-I)="SECTIONS REQ."
36 LET A$(M)="SECTION NUMBER."
38 LET A$(M+I)="(2 spaces) CARD NUMBER."
```

```
40  LET A$(L+M)="WORD OR CHR$."
42  LET A$(N-M-L)="(2 spaces) SETTING UP"
44  LET A$(N-M-I)="(3 spaces) CARD LIST"
46  LET A$(N-M)="(2 spaces) CARD SEARCH"
48  LET A$(L*M)="(2 spaces) CARD DELETE"
50  LET A$(N-M+L)="(1 space) CARD ENTRIES"
52  LET A$(N-L-L)="CARD ALTERATION"
54  LET A$(N-L-I)="(1 space) SECTION TRIM"
56  LET A$(N-L)="(1 space) S. (1 space) NAME CHANGE"
58  LET A$(N-I)="S. (1 space) ORDER CHANGE"
60  LET A$(N)="SECTION ENTRIES"
62  LET A$(N+I)="LIST ALL CARDS."
64  LET A$(N+L)="(2 spaces) T=TAPE OUT"
66  LET A$(N+L+I)="D/F=SLOW/FAST"
68  LET A$(N+L+L)="THIS FILE SET."
70  LET A$(N+M-L)="MAX.NO.OF CHR$."
72  LET A$(N+M-I)="REQ. (1 space) PER CARD."
74  LET A$(N+M)="**PLEASE WAIT**"
76  LET A$(N+M+I)="EXCEEDS NO. (1 space) OF"
78  LET A$(N+M+L)="PLEASE RE-ENTER"
80  LET A$(N+N/L)="J.E. (1 space) FILE CARD"
82  LET C=Q
84  LET D=Q
86  LET P=Q
88  LET Y=VAL "23"
90  LET Z$="4332122111"
92  LET C$="(shifted graphic 1)"
94  LET E$=C$
96  LET D$=""
98  LET H$="000102Y"
100 GOSUB T
102 LET F$=G$
104 LET H$="T001034"
106 GOSUB T
108 LET A=VAL G$
110 LET X=T*N/L/A
112 LET Y=Y+I
114 LET Z=Y+I
116 LET H$="X0010YZ"
118 GOSUB T
120 LET B=VAL G$
122 LET R=B
124 DIM B$(A,B)
126 GOSUB VAL "600"
128 GOSUB VAL "350"
130 CLS
132 PRINT TAB M+L;A$(N+N/L),,,A$(N+L+L);F$,,,A$(N-M+L,L TO ),D,,,
134 PRINT A$(N),C,,,A$(N+M-L);B,,,A$(L+L,L TO );A$(M-L, TO M-I);A,,,
136 PRINT "LENGTH OF (1 space)";A$(N),,,
138 FOR X=L TO LEN C$
140 PRINT CODE C$(X);":";
142 NEXT X
144 PRINT AT N,L+L;"(NEW-LINE TO CONTINUE.)"
146 GOSUB K
150 CLS
152 PRINT "OPTIONS" (Note: "Options" is in inverse CHR$)
154 FOR X=Q TO M+L
156 PRINT "(7 spaces)";X;"---";A$(X+N-M-I, VAL Z$(X+I) TO ),,,
158 NEXT X
160 PRINT A$(N+L),A$(N+L+I)
162 GOSUB VAL "700"
164 LET P=Q
166 GOSUB K
168 IF G$="" THEN GOTO VAL "130"
170 IF G$(I)>"." AND G$(I)<"A" THEN GOSUB VAL G$(I)* VAL "50"+T-K
172 IF G$="F" THEN FAST
174 IF G$="D" THEN SLOW
176 IF G$="T" THEN GOTO J*L
178 (This line is left blank for the sort option)
180 GOTO VAL "150"
200 LET H$="D411700"
202 GOSUB T
204 LET H$(I)="0"
206 LET W=VAL G$
208 CLS
210 LET P=N
212 PRINT A$(M+I);"- (1 space)";W;".",,,
214 FOR X=I TO C
216 GOSUB J
218 LET P=P-INT ((H-G)/VAL "32")
220 IF X>=P/(I+L) THEN GOSUB K
222 PRINT A$(N/L);TAB L+I;B$(W,G TO H)
224 PRINT
226 NEXT X
228 GOSUB K
230 RETURN
250 LET H$="C921600"
252 GOSUB T
254 FOR X=I TO VAL G$
256 GOSUB J
258 NEXT X
260 LET H$="0221800"
262 GOSUB T
264 DIM I$(I,H-G+I)
266 LET I$(I)=G$
268 CLS
270 FOR Y=I TO D
272 IF I$(I)<>B$(Y,G TO H) THEN LET P=P+I
274 IF I$(I)=B$(Y,G TO H) THEN PRINT Y;":";B$(Y, TO N+M+I)
276 IF Y=P THEN GOSUB K
278 NEXT Y
280 PRINT "(FINISH)"
282 GOSUB K
284 RETURN
300 LET H$="D431700"
302 GOSUB T
304 FOR X=VAL G$ TO D-I
306 LET B$(X)=B$(X+I)
308 NEXT X
310 LET B$(X)=""
312 LET R=R+I
314 LET D=D-I
316 RETURN
350 LET W=A-D
352 LET H$="W441034"
354 GOSUB T
356 FOR Y=I TO VAL G$
358 LET D=D+I
360 LET W=D
362 GOSUB VAL "700"
364 FOR X=I TO C
366 GOSUB J
368 LET H$="0041900"
370 GOSUB VAL "900"
372 IF LEN G$-CODE C$(X+I)>R THEN GOTO VAL "368"
374 IF CODE C$(X+I)>=LEN G$ THEN GOTO VAL "390"
376 FOR Z=I TO D
378 LET B$(Z)=B$(Z, TO H)+"(1 space)"+B$(Z,H+I TO )
380 NEXT Z
382 LET C$(X+I)=CHR$ (CODE C$(X+I)+I)
384 LET H=H+I
386 LET R=R-I
388 GOTO VAL "374"
390 LET B$(W,G TO H)=G$
392 NEXT X
394 NEXT Y
396 RETURN
400 LET H$="D451700"
402 GOSUB T
404 LET W=VAL G$
406 LET H$="C951600"
408 GOSUB T
410 FOR X=I TO VAL G$
412 GOSUB J
414 IF X=VAL G$ THEN GOTO VAL "368"
416 NEXT X
418 RETURN
450 LET H$="C961600"
452 GOSUB T
454 FOR X=I TO VAL G$
456 GOSUB J
458 NEXT X
```

```
46Ø  FOR Y=I TO D
462  LET BØ(Y)=BØ(Y, TO H-I)+BØ(Y,H+I TO )
464  NEXT Y
466  LET CØ(X)=CHRØ (CODE CØ(X)-I)
468  LET R=R+I
47Ø  RETURN
5ØØ  LET HØ="C9716ØØ"
5Ø2  GOSUB T
5Ø4  FOR X=I TO VAL GØ
5Ø6  GOSUB J
5Ø8  NEXT X
51Ø  LET X=X-I
512  LET HØ="Ø771Ø26"
514  GOSUB VAL "95Ø"
516  LET EØ(X+I)=CHRØ LEN GØ
518  LET DØ=DØ( TO E-I)+GØ+DØ(F+I TO )
52Ø  RETURN
55Ø  LET HØ="C9816ØØ"
552  GOSUB T
554  FOR X=I TO VAL GØ
556  GOSUB J
558  NEXT X
56Ø  FOR Y=I TO D
562  LET BØ(Y)=BØ(Y,G TO H)+BØ(Y, TO G-I)+BØ(Y,H+I TO )
564  NEXT Y
566  LET CØ="(shifted graphic 1)"+CØ(X)+CØ(L TO X-I)+CØ(X+I TO )
568  LET DØ=DØ(E TO F)+DØ( TO E-I)+DØ(F+I TO )
57Ø  LET EØ="(shifted graphic 1)"+EØ(X)+EØ(L TO X-I)+EØ(X+I TO )
572  RETURN
6ØØ  LET HØ="RØ91Ø35"
6Ø2  GOSUB T
6Ø4  LET Y=VAL GØ+C
6Ø6  FOR X=C+I TO Y
6Ø8  LET CØ=CØ+"(shifted graphic 1)"
61Ø  LET HØ="Ø991Ø26"
612  GOSUB VAL "95Ø"
614  LET EØ=EØ+CHRØ LEN GØ
616  LET DØ=DØ+GØ
618  LET R=R-I
62Ø  NEXT X
622  LET C=Y
624  RETURN
65Ø  FOR X=I TO C
652  GOSUB J
654  PRINT X;".";AØ(N/L),
656  IF X=P THEN GOSUB K
658  NEXT X
66Ø  LET P=P-X
662  PRINT AØ(N-M-L)
664  FOR Y=I TO D
666  GOSUB VAL "7ØØ"
668  IF Y=P THEN GOSUB K
67Ø  PRINT Y;":";
672  FOR X=I TO C
674  GOSUB VAL "716"
676  IF X+H+LEN STRØ Y>VAL "31" THEN GOTO VAL "682"
678  PRINT BØ(Y,G TO H);"(1 space)";
68Ø  NEXT X
682  PRINT BØ(Y,G TO VAL "31"-X-LEN STRØ Y) AND X<=C
684  NEXT Y
686  GOSUB K
688  RETURN
7ØØ  LET E=Q
7Ø2  LET F=Q
7Ø4  LET G=Q
7Ø6  LET H=Q
7Ø8  RETURN
71Ø  LET E=CODE EØ(X)+E
712  LET F=CODE EØ(X+I)+F
714  LET AØ(N/L)=DØ(E TO F)
716  LET G =CODE CØ(X)+G
718  LET H=CODE CØ(X+I)+H
72Ø  RETURN
8ØØ  INPUT GØ
8Ø2  LET P=P+N+L
8Ø4  CLS
8Ø6  RETURN
9ØØ  PRINT AT M-I,Q;AØ(M+I);W
9Ø2  GOTO T
95Ø  PRINT AT N-M-I,N+N/L;X
1ØØØ PRINT AT I,M;"(1 x shifted graphic E, 15 x shifted graphic 7,
     1 x shifted graphic R)";TAB M;"(1 x shifted graphic 5)";AØ(VAL
     HØ(L+I)+N-M-L);"(1 x shifted graphic 8)";TAB M;"(1 x shifted
     graphic W, 15 x shifted graphic 6, 1 x shifted graphic Q)"
1ØØ2 PRINT AT N/L,Q;AØ(VAL HØ(L+L)+I);AØ(VAL HØ(M-L)+I),,,AØ(VAL HØ
     (M-I)+I);AØ(VAL HØ(M)+I)
1ØØ4 INPUT GØ
1ØØ6 IF LEN GØ VAL>"255" THEN GOTO VAL "1Ø32"
1ØØ8 IF VAL HØ(I)<>Q THEN GOTO VAL "1Ø2Ø"
1Ø1Ø CLS
1Ø12 PRINT AT L,M+I;AØ(N+M)
1Ø14 RETURN
1Ø2Ø IF GØ="" THEN GOTO T
1Ø22 FOR Z=I TO LEN GØ
1Ø24 IF GØ(Z)<"Ø" OR GØ(Z)>"9" THEN GOTO T
1Ø26 NEXT Z
1Ø28 IF VAL HØ(I)<VAL GØ OR VAL GØ=Q THEN GOTO VAL "1Ø32"
1Ø3Ø GOTO VAL "1Ø12"
1Ø32 PRINT AT N-L,Q;AØ(N+M+I);AØ(VAL HØ(L)+N-M-L); AT N,M;AØ(N+M+L)
1Ø34 GOTO T
142Ø SAVE FØ
1422 GOTO VAL "13Ø"
```

Note that underlined instructions indicate the number of blank spaces or the graphic character symbol that should be entered in this position.

The program should now be run, tested and taped out, before continuing to add the sort option which follows. Then press Break whilst either the Option Page or the Report Page is being printed onto the screen (in Slow mode) in order to add the extra programming. Then add the following.

```
81 STOP
```

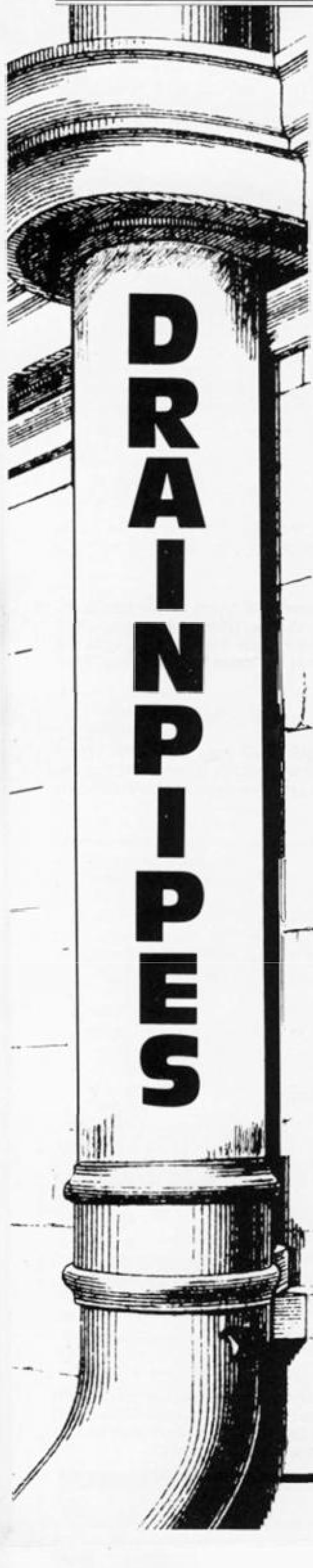
Then RUN the program. The program will stop at line 81. Now delete lines 8 to 81 inclusive (this can be done by entering the line numbers followed by Newline). The program must not be RUN again after this. Now add the following.

```
152  PRINT "OPTIONS" (Note:- "Options" is in inverse CHRØ),"(8 spaces)
     S=SORT"
178  IF GØ="S" THEN GOSUB T*L
2ØØØ FAST
2ØØ2 FOR P=I TO D
2ØØ4 IF L**P>D THEN GOTO VAL "2ØØ8"
2ØØ6 NEXT P
2ØØ8 LET W=L**P-I
2Ø1Ø LET W=INT (W/L)
2Ø12 IF W=Q THEN RETURN
2Ø14 LET X=D-W
2Ø16 LET Y=I
2Ø18 LET P=Y
2Ø2Ø LET Z=P+W
2Ø22 IF BØ(P)>BØ(Z) THEN GOTO VAL "2Ø3Ø"
2Ø24 LET Y=Y+I
2Ø26 IF Y>X THEN GOTO VAL "2Ø1Ø"
2Ø28 GOTO VAL "2Ø18"
2Ø3Ø LET XØ=BØ(P)
2Ø32 LET BØ(P)=BØ(Z)
2Ø34 LET BØ(Z)=XØ
2Ø36 LET P=P-W
2Ø38 IF P<I THEN GOTO VAL "2Ø24"
2Ø4Ø GOTO T*L+N
```

Now enter as a direct command:

GOTO 1

then press Newline. The program should now work complete with the alphanumeric sort option. To start a new file, always Break, then GOTO 1. If you have more than 16K of memory and wish to add more cards, then replace the 'X' in line 116 with a 'O'.



Here's an adaptation of an old arcade game written for us by Colin Gooch of Somerset.

Drainpipes is based on the old style of mechanical arcade game in which steel balls were dropped into cups or tubes.

The basis of the game is simple enough. A ball runs along a stepped roofline at the top of the screen and will drop down if you press 'D'. Below it are a series of seven 'drainpipes' and you score by making it drop into the centre of one of these. Each drainpipe will only score once and a red indicator will light up to show where a score has been made. A score in each pipe will give a maximum of 28 points.

This version of the game has a random 'flip' feature. A red and white flag will sometimes illuminate as the ball is dropping and at the same time, the rate of drop slows down. If you don't like where the ball is heading press 'K' and it will slide off to the left and make its way back to the start without incrementing the ball score. You can then try again. Alternatively, as it slides off to the left you can press 'D' again whereupon it will once more go on its downward path. After 12 balls (or a maximum score!) the computer will tell you your score and invite you to play again. If you don't want to continue, then pressing 'N' will get you a fond farewell.

Down the drain

The main movement of the ball and the setting of the graphics is done in lines 20 to 320. Lines 350 to 400 detect a scoring ball and label it with a score value. Lines 460 to 480 check to see if there is a full score line, ie that no tube still has a score of zero, and if necessary terminates the round by allowing the next loop 500 to 530 to add up the score. These lines also add the score if you run out of balls.

Lines 600 to 660 print out the score and start a new game. LInes 670 to 690 terminate the game. The 'flip' routine is in lines 710 to 890 and shift the ball around the screen as required. LInes 900 onward print out the game instructions.

Variations on a theme

Anyone who wishes can find a lot of variations that can be done with this program. For a simple game, you can omit the 'flip' routine by leaving out line 190 and lines 710 to 890. You can also omit line 15 and the instructions.

Line 100 looks a little clumsy, but it works! It could be done with a subroutine of print statements. It can also be made straight . . . though whether this makes for an easier or harder game is a matter of opinion.

You can make things a little more difficult by reducing 'ra' in line 190 thus getting less flips. You could also reduce the PAUSEs in lines 730 and 800 to increase the speed of each ball.

And if you really want to get your fingers confused on the keyboard, why not write an extra subroutine called by INKEY$ "L" which will cause the ball to slide to the right. If you do this, the subroutine would be called from line 740.

A Graphic explanation

To help you type the program in, Colin has provided us with an explanation of some of the lines.

Line 100 includes seven Graphics shifted 8s followed by 31 spaces, then six Graphics shifted 8s followed by 31 spaces.

Line 240 includes four Graphics shifted 8s, one Graphics shifted 5, one Graphics 8, one Graphics 5, one Graphics shifted 8, one Graphics shifted 5, and so on to until the end of the line which you should finish with a Graphics 5.

Line 360 includes one Graphics shifted 8.

Line 840 includes three Graphics shifted 8s.

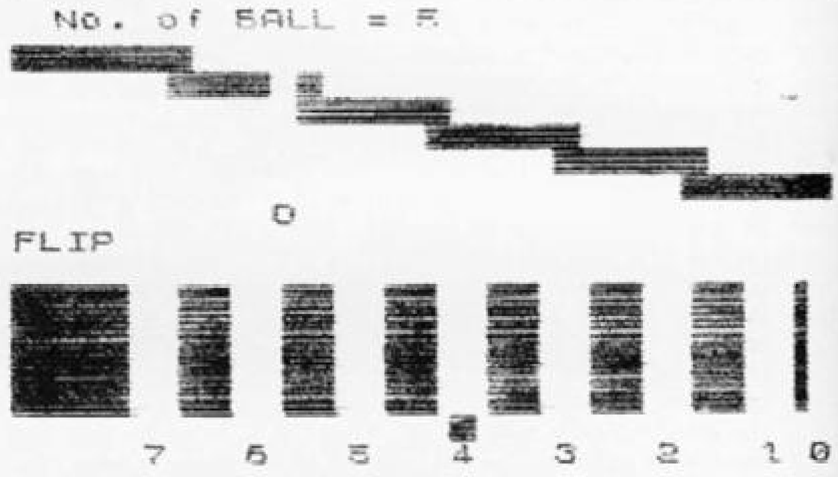
```
D R A I N P I P E S

 YOU WILL SEE A BALL ROLLLING
 ALONG A ROOFTOP.  YOU MUST GET
IT TO DROP INTO A DRAINPIPE BY
PRESSING "D"

 YOU CAN ONLY REGISTER A SCORE
ONCE IN A PIPE, A RED INDICATOR
SHOWS WHERE THERE HAS BEEN A
 SCORE

 IF A FLAG FLIP SHOWS ON THE
LEFT PRESSING "K" WILL CAUSE
 THE BALL TO MOVE TO THE LEFT

 THE BALL WILL RETURN TO THE
START UNLESS YOU PRESS "D" AGAIN
   IF YOU DO THE BALL WILL DROP
IN THE NEW PLACE
```




```
 5 REM DRAINPIPES
10 REM BY C.N.GOOCH ©1982
15 CLS : GO SUB 900
20 LET ps=0: LET pb=12
```

```
 30 BORDER 5: PAPER 6: CLS : LE
T sc=0
 40 LET ba=0: LET bb=0
 50 DIM c(7)
 60 LET p=2
 80 PRINT AT 20,0; PAPER 4;"
 7   6   5   4   3   2   1 0"
100 LET a$="████▄▄▄▄▄▄████▄▄▄▄▄▄
▀▀▀▀▀████▀▀▀▀▀▀▀▀▀████"
 110 PRINT AT 3,0; INK 2;a$
 120 GO SUB 240: REM SET UP ROOF
 130 LET a=0: LET b=2: PAUSE p*5
0
 140 PRINT AT b,a; INK 3;" O": P
RINT AT b-1,a;" ": IF a=0 THEN B
EEP .15,-20: REM START BALL
 150 LET ra=1+INT (RND*10): LET
a=a+1: IF a=7 OR a=12 OR a=17 OR
 a=22 OR a=27 THEN LET b=b+1
 160 IF a=31 THEN GO TO 190
 170 IF a>3 AND INKEY$="d" OR A>
3 AND INKEY$="D" THEN BEEP .15,1
0: GO TO 190
 180 GO TO 140
 190 IF ra<4 THEN GO TO 710: REM
 DROP BALL
 200 PRINT AT b,a; INK 3;"O": PR
INT AT b-1,a;" ": LET b=b+1
 210 IF b=19 THEN LET b=19: GO S
UB 350: GO TO 230
 220 GO TO 190
 230 LET p=1+INT (RND*3): GO SUB
 250: GO TO 130
 240 LET b$="████ ██ ██ ██ █
██ ██ ██ █ "
 250 FOR n=1 TO 5: REM SET UP PI
PES
 260 PRINT AT 13+n,0; INK 1;b$
 270 NEXT n
 290 PRINT AT 3,0; INK 2;a$
 300 LET ba=ba+1: PRINT AT 0,2;"
No. of BALL = ";ba: IF ba=13 THE
N GO SUB 500: PAUSE 100: GO TO 6
00
 310 GO SUB 460
 320 IF sc>=28 THEN PRINT AT 0,1
8;"Score= ";sc;AT 1,8; INK 2; PA
PER 7; FLASH 1;"M A X I M U M ":
 PAUSE 200: GO TO 600
 330 RETURN
 350 FOR i=5 TO 29 STEP 4: REM I
NDICATE SCORE
 360 IF b=19 AND a=i THEN PRINT
AT 19,i; INK 2; FLASH 1;"█"
 370 IF a=i AND c((33-i)/4)=0 TH
EN BEEP .15,50
 380 IF a=i THEN LET c((33-i)/4)
=(33-i)/4
 390 NEXT i
 400 RETURN
 460 FOR q=1 TO 7: REM DETECT FU
LL SCORELINE
 470 IF c(q)=0 THEN RETURN
 480 NEXT q
 500 FOR l=1 TO 7: REM ADD SCORE
S
 510 LET sc=sc+c(l)
 520 NEXT l
 530 RETURN
 600 CLS : PRINT AT 3,5; INK 1;
PAPER 6;" YOUR SCORE WAS ";sc;AT
 5,7;" IN ";(ba-1);" BALLS "
 610 IF (ba-1)<pb THEN LET pb=(b
a-1)
 620 IF sc>ps THEN LET ps=sc
 625 PRINT AT 6,4; PAPER 1; INK
7;" HIGHEST SCORE SO FAR ";ps;AT
 7,10;"IN ";pb;" BALLS "
 630 PRINT PAPER 4;AT 12,0;"...D
O YOU WANT ANOTHER GO?      ..."
  PRESS", PAPER 7, INK 2;"Y"
; PAPER 4; INK 0;"ES OR "; PAPER
 7; INK 2;"N"; PAPER 4; INK 0;"O
"
 640 IF INKEY$="Y" OR INKEY$="y"
THEN GO TO 25
 650 IF INKEY$="n" OR INKEY$="N"
THEN GO TO 670
 660 GO TO 640
 670 CLS
 675 PRINT TAB bb+5; PAPER 1+INT
 (RND*7); INK 9; BRIGHT 2-(1+INT
 (RND*2));" THANKS FOR PLAYING,
BYEE E  E  E   E ": IF bb>55 THE
N GO TO 690
 680 LET bb=bb+1: POKE 23692.255
: BEEP .25,bb: GO TO 675
 690 CLS : PRINT AT 4,4; PAPER 6
; INK 2: BRIGHT 1: FLASH 1:"  A
N Y O N E  W H N T  A  G A M E
 ?  "'TAB 4;"  PRESS ENTER TO ST
ART  ";AT 15,7; INVERSE 1;"D R A
 I N P I P E ": INPUT Q$: IF Q$=
"" THEN GO TO 25
 710 IF b<13 THEN PRINT AT 10,0;
 INK 2; PAPER 7; FLASH 1; BRIGHT
 1;"FLIP"
 720 IF b=13 THEN PRINT AT 10.0;
 PAPER 6; FLASH 0; BRIGHT 0;"
 "
 730 PRINT AT b,a; INK 3;"O";AT
b-1,a;" ": PAUSE 5
 740 IF b>10 AND b<14 AND  INKEY
$="k" OR INKEY$="K" THEN GO TO 8
00
 760 LET b=b+1
 770 IF b>=14 THEN GO TO 210
 780 GO TO 710
 800 PRINT AT b,a; INK 3;"O ": P
AUSE 2
 810 LET a=a-1: IF a=2 THEN PRIN
T AT 10,0;"    ": IF a=2 THEN GO
 TO 830
 815 IF INKEY$="d" OR INKEY$="D"
 THEN GO TO 870
 820 GO TO 800
 830 PRINT AT b,a; INK 3;"O ";AT
 b+1,a;" ": IF b=12 THEN PRINT A
T 14,0; INK 1;b$
 840 LET b=b-1: IF b=2 THEN PRIN
T AT 10,0;"     ";AT 3,2; INK 2;"
██"
 850 IF b=2 THEN GO TO 140
 860 GO TO 830
 870 PRINT AT b,a; INK 3;"O ";AT
 b-1,a;" ";AT 10,0;"
 880 LET b=b+1: IF b=19 THEN LET
 b=19: GO SUB 350: GO TO 230
 890 GO TO 870
 900 BORDER 6: PRINT PAPER 5; IN
K 2;AT 1,5;" D R A I N P I P E S
 "
 920 PRINT PAPER 7;AT 3,0;"  YOU
 WILL SEE A BALL ROLLLING  ALON
G A ROOFTOP.  YOU MUST GET IT TO
 DROP INTO A DRAINPIPE BY  PRESS
ING ""D"""
 930 PRINT AT 8,0; PAPER 7;" YOU
 CAN ONLY REGISTER A SCORE  ONCE
 IN A PIPE, A RED INDICATOR SHOW
S WHERE THERE HAS BEEN A    SCO
RE                              "
 940 PRINT AT 13,0; INVERSE 1;"
IF A FLAG "; INK 2;" FLIP "; PAP
ER 7; INK 0;"SHOWS ON THE   LEFT
  PRESSING ""K"" WILL CAUSE
THE BALL TO MOVE TO THE LEFT   "
 950 PRINT AT 17,0; PAPER 7;" TH
E BALL WILL RETURN TO THE    STA
RT UNLESS YOU PRESS ""D"" AGAIN
  IF YOU DO THE BALL WILL DROP I
N THE NEW PLACE                 "
 960 INPUT "PRESS ENTER TO START
"; LINE Z$: GO TO 20
```

NOW FROM AFDEC
ZX81

ZX-PANDA

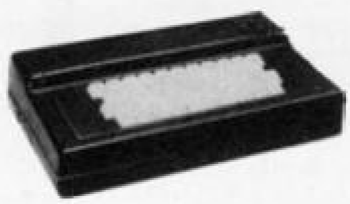


16K Expandable RAM

Uniquely expandable to 32K by the addition of our 16K Expansion Module.

ZX-PANDA Basic 16K RAM Pack	£25.00 incl
ZX-PANDA 16K Expansion Module	£19.95 incl
GIANT PANDA Full 32K RAM Pack	£40.00 incl
Finally the Really big One 64K RAM Pack	£65.00 incl

KEYBOARDS
TYPE A CASED 42 KEY

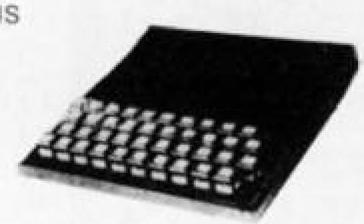


£30.00 Incl

The ZX81 fits inside
The tough ABS injection moulded plastic case measures 8" × 14" × 2½" and hooks up to your ZX printed circuit board in minutes. No technical know how or soldering is required.

TYPE B 41 KEY REPLACEMENT

The famous **'KLIK'** Keyboard



£26.50 incl

- Fits in the Recess left after peeling off existing touch sensitive Keypad.
- No Soldering • No trailing wires.
- Positive feel and sound as key operates.

Cheques/PO please to

AFDEC ELECTRONICS LTD
318 Kempshott Lane, Basingstoke, Hants RG22 5LT

PLEASE SUPPLY (all prices include VAT & P&P)

16K EXPANDABLE RAM	£25.00
16K EXPANSION MODULE for above	£19.95
32K GIANT PANDA	£40.00
64K THE BIG ONE	£65.00
KEYBOARD TYPE A	£30.00
KEYBOARD TYPE B	£26.50

PLEASE PRINT CLEARLY

NAME ____________________

ADDRESS ____________________

HORSE RACING

ZX81

16K. This programme analyses 17 separate inputs of information on each horse's past performance to forecast winners. The programme comments on the important factors such as the horse's racing record, weight etc. etc. Also all the information is assessed and correlated to a numerical value. Full instructions and a 14 day 50% money back guarantee if not fully satisfied. **£12.00**

1K. This smaller programme also analyses all the horses in a race and will list the most likely winners. On the first day ever trated it picked 8 winners from only 8 races (20.3.82). Full instructions included and a 14 day 50% money back guarantee if not fully satisfied. **£8.00**

FORTUNE TELLING

The predictions in this full 16K programme will amaze you and your friends. Covers your future, day to day gambling etc. etc. A 14 day 50% money back guarantee if not fully satisfied. **£5.00**

Send cheque or P.O. to the

BUTRONICS CO.
44-46 Earls Court Road,
London W8 6EJ.

(Mail order only please)

ZX LOADING AID for

ZX80, ZX81 or ZX Spectrum [Please specify]. Removes guesswork from adjusting volume. Plugs in between computer and tape recorder. Red & green LEDs on — Optimum Volume Set. Red LED [only] on — volume too low. Yellow LED on — Volume too high.

£11.95 [incl. instructions, p&p & VAT]

Push button to reset 'K' Cursor **£1 extra**
Earphone & SKT to monitor 'Voice Overs' **£1.50 extra**

ZX81 KEYBOARD BLEEPER

Provides feedback missing from touch sensitive keyboard cheaply — easy installation. All 210 characters bleep in slow & fast modes. No soldering required — all connections plug-in. Illustrated instructions supplied. Fits inside case either under keyboard or ZX81 PCB. No trailing wires — also suits most full size keyboards.

£9.95 [incl. illustrated instructions, p&p & VAT]

On/Off Switch **£1.50 extra**

T-SHIRTS £3.50 SWEATSHIRTS £7.50

Black with Red "SINCLAIR ZX81" or White with Black "ZX SPECTRUM" + Rainbow.
Terrific hand airbrushed multicoloured designs. "I'm a Micro Addict" or "Micro Computers Take You Into Another World", against space scene.

T/S £4.50 S/S £9

Specify size required — 24" — 44"

SEND SAE FOR FURTHER DETAILS

FULCRUM PRODUCTS, Dept B,
Hillside, Steep Lane, Findon, West Sussex BN14 0UF.

Proctim

If you're still in the dark about using your ZX81 as a process timer, Peter Coupe shows you how.

The object of this program is to enable me to use my ZX81 as a process timer in the photographic darkroom. Most photographic processes involve a number of steps and all these steps can have different times. Keeping track of which step in a process you are in the middle of, and how long it should last, makes it difficult if not impossible to do anything else at the same time as processing work. Obviously a device which keeps track of these things for you — and gives an instantly 'checkable' graphic readout — is going to make working much easier (and more foolproof!). Timers are available commercially, but many of these units will cost almost the same as a new ZX81 — and you can't play space invaders on any commercially produced timer that I know of!

The program listing is fairly straightforward and it involves no concepts beyond the scope of the average ZX81 programmer.

RUNning the program

The program splits, basically, into three major areas:

a. Input sequence.
b. Timing sequence.
c. Graphics display.

At the start of the RUN, you will be presented with the word STEPS. Here you input a number which corresponds to the number of steps in the sequence that you wish to time, between 1 and 4. Next, you will be prompted to give a name, of up to four letters, to step 1. Then, you will be asked for a time, from one minute to 10 minutes in 15 second blocks, for step 1. This sequence of naming and timing will continue until all steps are done. When all are done you will be given N/L. When you press Newline, the sequence will start.

The screen will now clear and the name of the first process and the time that you have allocated to it will appear on the top line. A vertical plot will also appear.

The vertical plot is proportional in size to the length of time that the step will take. As 15 second intervals go by, one pixel at a time is nibbled away from the original vertical plot. The bottom pixel is left to show the start of the plot. When there are only two pixels left, at the top and bottom, you are into the last 15 seconds for this particular step. At the end of the step the screen clears, a new top line appears and a new plot appears. At the end of the sequence, the word END appears.

The program runs happily in my unexpanded ZX81. Obviously, with more memory, a much more elegant program could be produced using Proctim as its base. I have found the program most useful in a number of applications in the processing of both film and paper where daylight containers are used. I suppose with the right colour filter over the VDU it could also be used in true 'darkroom' situations.

```
4    DIM Z(4)
5    DIM N$ (4,4)
10   PRINT "STEPS"
20   INPUT A
30   GOSUB 500
45   PRINT "N/L"
46   INPUT F$
47   CLS
50   GOSUB 900
100  PRINT "END"
101  STOP
500  FOR K=1 TO A
510  PRINT "NAME STEP";K
520  INPUT K$
530  LET N$(K)=K$
540  PRINT "TIME FOR";
     N$(K)
720  INPUT Z
730  LET Z(K)=Z
740  NEXT K
750  RETURN
900  FOR P=1 TO A
910  PRINT N$(P);" FOR";
     Z(P);" MINS."
915  LET W=(Z(P) * 4)
920  FOR R=0 TO W
930  PLOT 0,R
940  NEXT R
950  FOR Q=1 TO W
955  PAUSE 748
960  UNPLOT 0,Q
980  NEXT Q
985  CLS
990  NEXT P
999  RETURN
```



Memory remaining

```
9999 PRINT  "MEMORY ALLOCATION",,,
            "PROGRAM", USR 16514,
            "DISPLAY FILE", USR 16522,
            "VARIABLES", USR 16531,
            "WORK SPACE", USR 16540,
            "STACKS", USR 16559,
            "SYSTEM VARS", "125",,,
            "SPARE", USR 16549
```

The BASIC part of the listing.

If you've ever wondered what your computer does with all its memory, Ian Turtle has devised a useful program to tell you.

This program is a 'memory remaining/memory expanded' program with a difference. Instead of simply stating '7193 bytes used' on the screen, it splits up how the computer has used the bytes and how many bytes are remaining for future use. For example, on my 8K ROM ZX80 with 8K RAM, an output may be as follows.

MEMORY ALLOCATION

PROGRAM	2943
DISPLAY FILE	793
VARIABLES	48
WORK SPACE	0
STACKS	18
SYTSTEM VARIABLES	125
SPARE	4265

It should be noted that the total number of bytes displayed adds up to 8192 which is 8 * 1024, ie the total RAM available on an 8K computer.

Thanks for the memory

The actual program itself contains a REM statement with 61 bytes of machine code, and another line of BASIC to call the machine code and present the display as shown above. The machine code segment consists of six small subroutines which are called from the line of BASIC.

The titles on the left of the screen may be noticed to be roughly the same as the system variable titles on pages 177-179 of the Sinclair manual. This is how the program actually works, selecting the required areas to be calculated from the system variables.

The machine code is meant to put in line 1 with a REM statement. Although the code is completely relocatable, the BASIC line (line 9999) assumes a REM statement when calling.

Should you not wish to use a Hex loader, the machine code can be loaded using the following program.



```
1 REM (containing 61 of any
  character)
2 FOR I = 16514 TO
  16574
3 INPUT A
4 POKE I, A
5 SCROLL
6 PRINT PEEK I,I
7 NEXT I
```

RUN this program and input the decimal values given in the machine code listing. Note any mistakes made as these must be 're-POKEd' before the end. When the last value (201) has been input, the program (if everything has been input correctly — or at least the correct number of figures have been input) will terminate with 0/7.

If found useful, the REM statement and line 9999 which calls the machine code can be SAVEd on tape and re-LOADed before inputting any program. The 'SPARE' is very useful as it give the bytes remaining at any time.

Any RAM to spare?

If the user requires the spare RAM option without the rest, the program in Table 1 will achieve this.

Op-codes	Hex	Decimal	Comments
LD HL,0000	210000	33,0,0	Clear HL register pair
ADD HL,SP	39	57	Set HL to value of SP
LD BC, (STKEND)	ED4BIC40	237,75,28,64	Ld BC with STKEND
ADD A,0	C600	198,0	Clear carry
SBC HL,BC	ED42	237,66	Find memory left
LD B,H	44	68	Put valve into BC
LD C,L	4D	77	for convenience
RET	C9	201	Return to BASIC

This machine code can be placed anywhere, though perhaps it would be best in a REM statement in line 1 (which will require 15 characters). Using PRINT USR (address) will print out the memory remaining.

◀ *Table 1.*



PROG	LD,HL,(D-FILE)	2A0C40	42,12,64	Program
	LD BC,16509	017D40	1,125,64	
	JR CALC(+46)	182E	24,46	
DIS-F	LD HL,(VARS)	2A1040	42,16,64	Display File
	LD BC,(D-FILE)	ED4B0C40	237,75,12,64	
	JR CALC(+37)	1825	24,37	
VA-BLES	LD HL,(E-LINE)	2A1440	42,20,64	Variables
	LD BC,(VARS)	ED4B1040	237,75,16,64	
	JR CALC(+28)	181C	24,28	
SPACE	LD HL,(STKEND)	2AIC40	42,28,64	Calculator Stack, Line being typed + work space
	LD BC, (E-LINE)	ED4B1440	237,75,20,64	
	JR CALC(+19)	1813	24,19	
SPARE	LD HL,0000	210000	33,0,0	Spare RAM
	ADD HL,SP	39	57	
	LD BC,(STKEND)	ED4BIC40	237,75,28,64	
	JR CALC(+9)	1809	24,9	
STACK	LD HL,0000	210000	33,0,0	Machine Stack + GOSUB Stack
	ADD HL,SP	39	57,	
	PUSH HL	ES	229	
	POP BC	CI	193	
	LD HL,(RAMTOP)	2A0440	42,4,64	
CALC	ADD A,0	C600	198,0	Calculate figure
	SBC HL,BC	ED42	237,66	
	LD B,H	44	68	
	LD C,L	4D	77	
	RET	C9	201	

The machine code part of the listing.

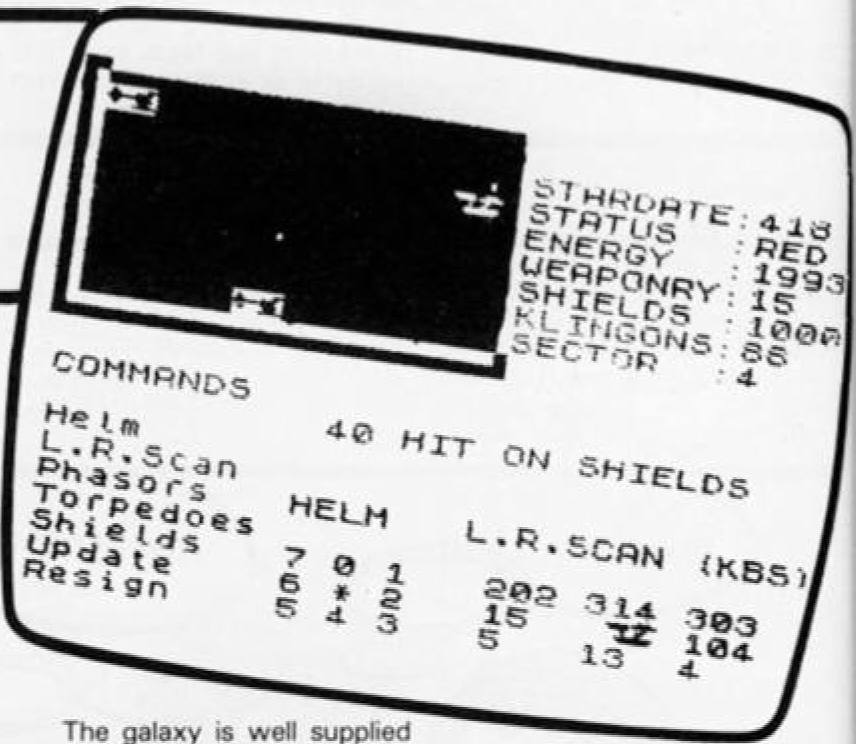
On your Spectrum

Phil Garratt takes a long, hard look at some of the software available for the ZX Spectrum.

Startrek — Fuller Micro Systems

Fuller Micro Systems' Startrek is a no-frills version of the classic game, for the 16K Spectrum. Your task, as ever, is to clear the eight by eight sector galaxy of the cursed Klingons. Nicely detailed user-defined graphics are used to mark the Enterprise and any Klingons or starbases in your current sector. You are also told the state of your energy reserves, shields and torpedoes. Sectors are numbered consecutively, rather than in the more usual line/column format.

Seven commands are available in this version. 'Helm' requires a direction from 0 to 7 and 'Warp Factor' from 1 to 63, and is used for movement. 'Torpedo' uses the same direction input, and will always destroy any Klingon it hits, whereas 'Phasers' seem to need an awful lot of energy to knock out the enemy. 'Update' is equivalent to a short range display, and has to be used if you want the destroyed Klingon to disappear from the screen. Updating is done automatically when the Enterprise moves. The 'Long Range Scan' shows Klingons, starbases and stars in the surrounding eight sectors and is slightly confusing as it is not justified, so that if there are only stars in a sector, it appears in the Klingons column. 'Shields' set the strength of your defences and the last command is 'Resign' for when you tire of the chase.



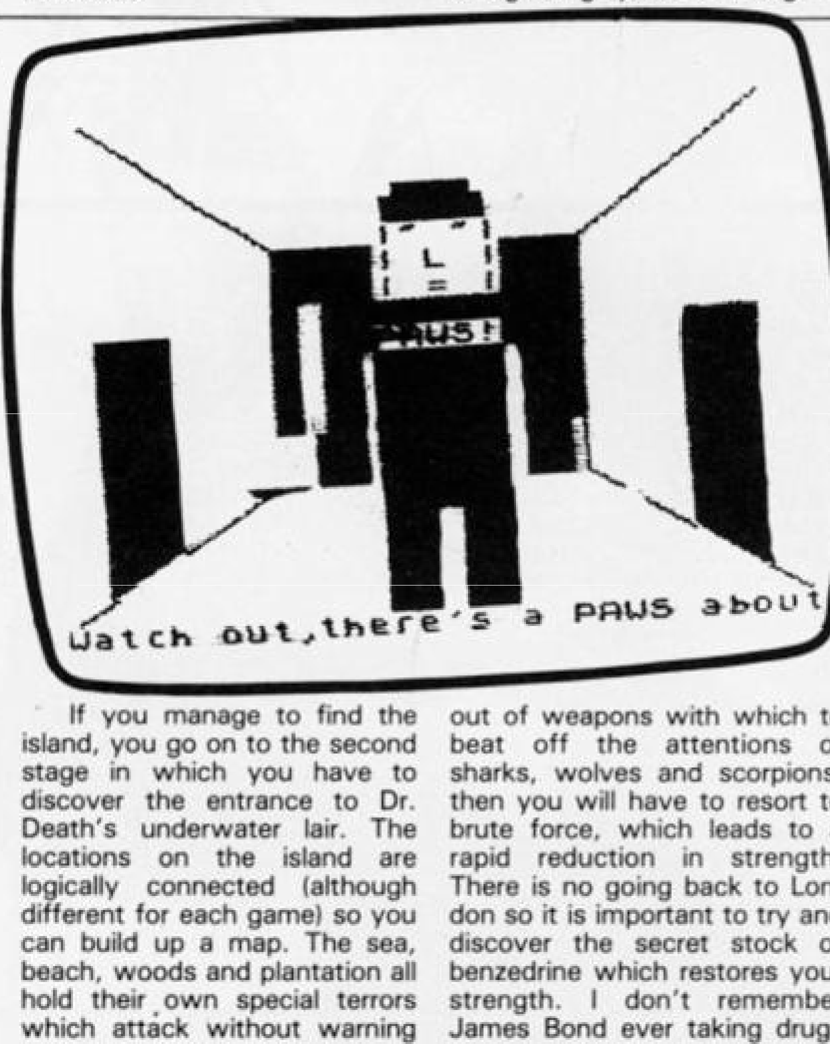
The galaxy is well supplied with starbases at which you can refuel, but I can't help feeling that with such a limited number of commands and scenarious, you'd have to be a pretty dedicated 'Trekkie' to see this one through to the bitter end. However, it is reasonably quick, has good graphics (although it would have been nice to see the torpedo tracks) and sound effects.

'Startrek' is priced at £5.50 and is available from Fuller Micro Systems, The ZX Centre, Sweeting Street, Liverpool 2.

Shaken but not stirred! — Richard Shepherd Software

The 48K Spectrum rather than the cinema is the venue for the latest James Bond epic. In Richard Shepherd Software's 'Shaken but not stirred' you play the part of 007 sent on a dangerous mission to disarm a nuclear missile, which has been stolen by the dastardly Dr. Death. Having been briefed by M and after selecting your weapons from the range that Q has to offer, you set out on your adventure.

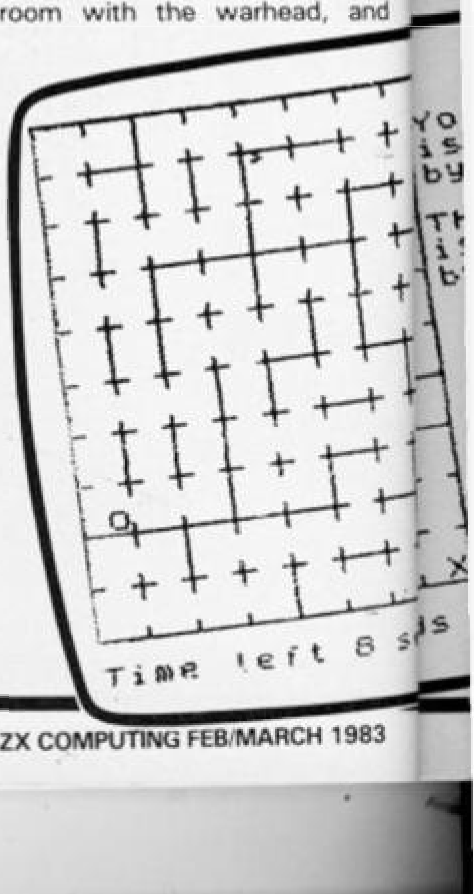
In the first stage, you must travel the world and try to stay alive long enough to gather sufficient clues to identify the island on which Dr. Death has his base. Muggers, midgets and priests are out to get you, and you will also have to cope with offers of mysterious meetings, unidentified packages and suspicious bowls of fruit. As long as you don't do it too often, you can return to London to replenish your strength and restock your personal armoury.



If you manage to find the island, you go on to the second stage in which you have to discover the entrance to Dr. Death's underwater lair. The locations on the island are logically connected (although different for each game) so you can build up a map. The sea, beach, woods and plantation all hold their own special terrors which attack without warning as you move around. If you run out of weapons with which to beat off the attentions of sharks, wolves and scorpions, then you will have to resort to brute force, which leads to a rapid reduction in strength. There is no going back to London so it is important to try and discover the secret stock of benzedrine which restores your strength. I don't remember James Bond ever taking drugs (apart from the occasional Mickey Finn!) and I'm not sure it's the sort of idea that should be incorporated in a game.

For your eyes only

I never managed to find the lair, but if I had I would have been confronted with a 10 by 10 room maze, displayed three dimensionally on the screen. Somewhere in it is the control room with the warhead, and



also Paws, the steel fisted baddie who is too strong to fight against, and so must be avoided at all costs. Movement is by using the cursor controls, and you can also get a map of the maze, but this is displayed for just 10 seconds and can only be used three times. If you find the control room, you then face a mastermind-type puzzle to crack the secret code which defuses the bomb. And if you manage that then I think you deserve a vodka martini – shaken not stirred, of course!

This is the best program I've seen from Richard Shepherd, and there is certainly plenty of it. The game can be SAVEd, but once restored cannot be SAVEd again. I think it's a shame that all the graphics are left for the last section, which is very hard to reach. Little use is made of sound and rather too much use is made of the RND function for me to be hooked.

'Shaken but not stirred' is priced £5.50 and is available from Richard Shepherd Software, Freepost, Maidenhead, Berks, SL6 5BY.





Football manager – Addictive Games

Versions of Addictive Games' Football manager have been produced for all the popular home micros, and it is now available for the 48K Spectrum.

You are given the managership of any one of the 64 teams in the league (you can even change any team's name to your own favourite if it is not one of those shown). Whichever team you choose, it begins in division four at the start of a new season. Your aim is to achieve promotion and a good run in the FA Cup – who knows, perhaps even win it!

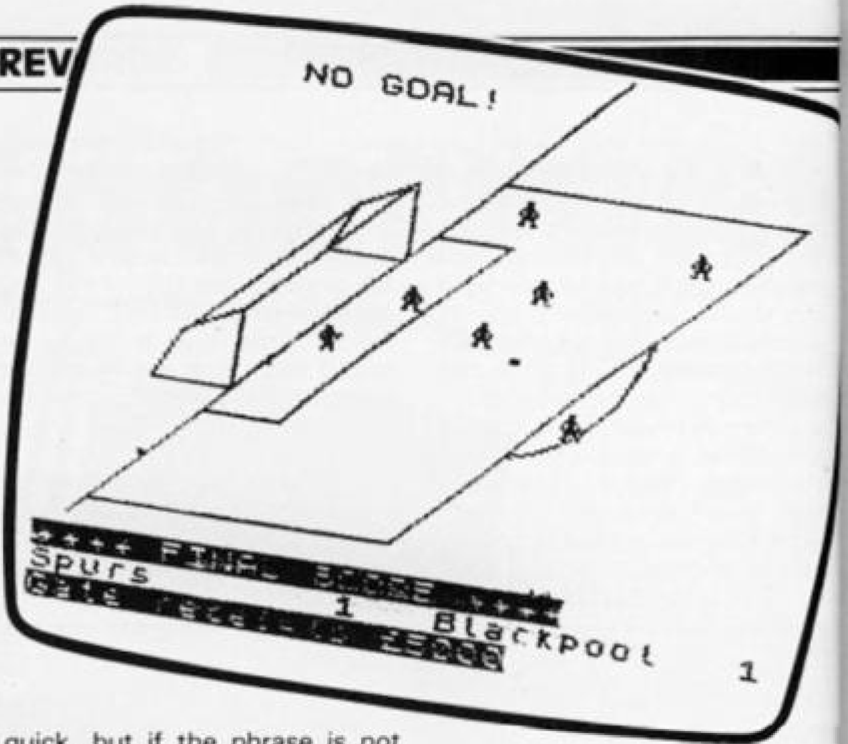
The league is split into four divisions of 16 teams, so each season involves 15 league matches plus up to eight rounds of the FA Cup. You start off with a squad of 12 players and £100,000 to spend in the transfer market if you wish. If you're the sort of manager who likes to try and buy your way to the top, then you can also obtain a bank loan up to a certain credit limit based on your division, at a 1% per week rate of interest.

The players, whose names are those of current league footballers, have three attributes. First, they are either defence, midfield or attacking players. Second, they have a skill rating of 1 to 5, on which depends their value. Third, they have an energy rating of 1 to 20, which goes down by one for each game played, and up by 10 for each game rested. When you come to play a match you are given the 'team attributes' for yourself and the opposition, and it is then up to you to juggle with the composition of your team if you need to. The five team attributes are the average energy rating, morale (which goes up and down depending on results), and then the total skill rating in defence, midfield and attack.

With the team selected, you can sit back and watch the match highlights in moving 3-D graphics. This is what makes the game more than just a sophisticated 'Kingdoms'. Seven or eight goalmouth incidents are shown, with players running around, moving into position, and then shooting at goal. Having to helplessly watch the results of your decisions like this is almost as exhausting as playing. The final score is based on the team attributes, but there is always the chance of a shock result. After the highlights, if it was a league match the rest of the results are given and the league table calculated and displayed.

Match of the day

Each week you are shown your financial balance sheet. Outgoings are wages, which depend on the number of players and their value, ground rent, and interest on your loan (if any). Incomings are gate receipts, which are based on your position and your opposition's position in the league. A good FA Cup run can also be a money spinner. At the end of the season, promotion and relegation take place, you are given a bonus according to your league ▶



position and a 'managerial success rating' is calculated.

There is a facility to SAVE the game at any point so that you can progress with one team for as long as you like. This feature is often available in adventure or chess programs, but I always find I forget where I've been, or what strategy I was following. With Football Manager, this is not a problem, as all the information you need is available at each stage.

Although I'm no great football fan, I really enjoyed playing this game. Despite having been converted from a ZX81 program, excellent use is made of colour and user-defined graphics. The game is very logically put together, so that the development of strategy and tactics has a real effect. For example, one of my teams got through to the fourth round of the FA Cup where it was beaten by a second division side. This upset morale and meant that our promotion bid failed. Perhaps I should have given up the FA Cup run and held some good players back – the possibilities are endless. Brian Clough had better watch out!

'Football manager' is £7.95 and is available from Addictive Games, PO Box 278, Conniburrow, Milton Keynes, MK14 7NE



Pimania — Automata

I thought Pimania was the reason I'm two stone overweight until I received Automania's new adventure program for the 48K Spectrum. Advertised as 'The Adventure Game that's for Real', there is more than just satisfaction awaiting the person who cracks this puzzle. If you interpret the clues correctly you can work out the time, date and place where someone will be waiting to hand over the 'Golden Sundial of Pi'. The Sundial is £6,000 worth of gold, diamond, lapis lazuli and obsidian, crafted by the award-winning designer, Barbara Tipple. The prize is on show at Southsea, and will also be displayed at computer fairs and exhibitions (accompanied by Securicor, I hope, in case anyone thinks of a less subtle method of winning it!).

The program starts in a none-too-friendly way with dire threats against anyone attempting to pirate Pimania. Then you have to work out the 'key' which unlocks the First Gate of Pi, and you get to meet the Pi Man for the first time. This all-singing, all-dancing little creature appears from time to time in the adventure, and may help or hinder your progress.

Baked beans?

The adventure itself is not a particularly large one, around 20-30 locations, with only short descriptions. The locations are logically connected, and you move between them by entering a number, which may or may not have been suggested in the description. Objects are randomly scattered around, and a pretty odd bunch of objects they are – a cross between the 'Generation Game' conveyor belt and the adverts that come after. A cuddly toy, hula hoop, pork pie, baked beans and quite a few others are to be found, although I didn't have time to do anything very useful with them afterwards!

The program's vocabulary is described as 'absolutely vast', but if you exclude the objects and words which have no useful effect, the actual vocabulary is pretty small. The processing of words entered is reasonably quick, but if the phrase is not understood or a number is entered which is not a valid exit, the command is rejected and the location re-displayed on the screen. This makes the game rather slow to play.

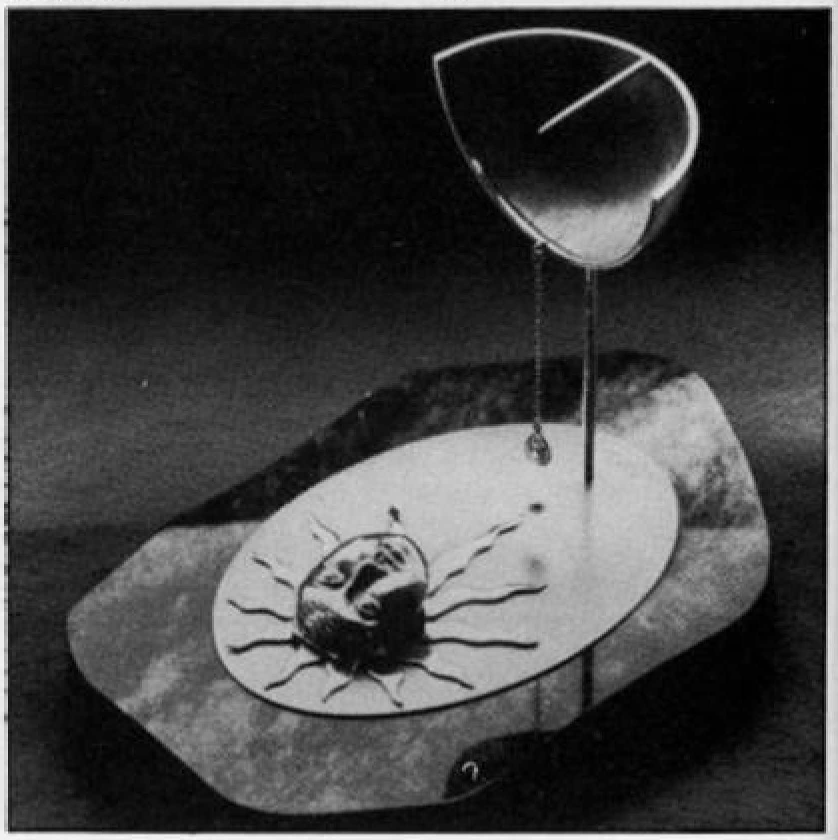
Automata are (in)famous for their ZX81 'cans of worms' programs, and despite the attempt to go more up-market with Pimania, some of the program is distinctly seedy. Rancid sewage pipes flow into clogged canals, watch out for the scab-infested odorous pit and you can guess what the sound effects are when you collect the baked beans! If frustration causes you to use language more suitable for the 'can of worms', you will be harshly punished by the Pi Man.

'Pimania' is £10 and is available from Automata Ltd., 65A Osborne Road, Portsmouth PO5 3LR.

All singing, all dancing

Where the program does score is in the large number of tunes and the clever use of moving user-defined graphics. The tunes range from the 'Hokey Cokey' complete with dancing Pi Man, to a gratingly not-quite-right rendering of 'Lucy in the Sky with Diamonds' when you collect the valium!

I didn't have sufficient time to work out how the objects relate to each other or to the locations, so I haven't any idea what the solution is or what form it might take. But with so many objects, tunes, locations and graphics, some or all of which may provide clues to the treasure, there's certainly many hours of detective work to put in.



House names



Should you have a problem searching for a house with a name but no number, Mr Graham of County Durham has come up with a clever solution.

This program, written for the ZX81 with 16K RAM Pack, following initial entry will start immediately with a main menu. This menu offers you seven options: a list of house names; a search for a house name; a display of a street; the opportunity to enter or amend a house name; to start a new file; to save a file on tape; or to finish.

Ten maps of 20 lines can be entered line by line and after 10 lines have been entered you are asked if you would like to correct any errors. If you make an error at this stage there is no need to worry as you can always amend your entries later using option 4.

House Names will search for a property using the first few letters of the name and then display a map on the screen with a flashing cursor in inverse video adjacent to the house in question. On request, the computer will then search for another property. Once you have the correct display on screen, you can copy the screen onto a printer.

Save and finish routines are included in the listing. For a more detailed guide to the listing, here is a table of the variables used in the program.

B	– Map screen number.
C	– Map line number.
D	– Flag check.
E	– Loop counter.
F	– Loop counter.
G	– Flashing cursor line number.
H	– Flashing cursor loop counter.
S	– Save flag check.
X	– Temporary map screen number.
M$	– Map string.
N$	– Name string.
P$	– Temporary test.
T$	– Test string.
X$	– Input check.
Z$	– Input.

The reason why so many variables were used was to protect the screen maps. Should the program crash, try typing GOTO MENU and restart from there.

```
   1 FAST
   2 CLEAR
   3 DIM M$(10,20,10)
   4 DIM N$(10,20,16)
   5 DIM R$(200,20)
   6 DIM Z$(1)
   7 DIM X$(2)
  50 LET B=1
  51 LET C=1
  52 LET D=0
  53 LET E=1
  54 LET J=0
  55 LET N=1
  56 LET MENU=200
 100 REM **S.GRAHAM. **
 200 CLS
 210 PRINT TAB 6;"**HOUSE NAME F
ILE**"
 215 PRINT
 220 PRINT "YOU HAVE THE FOLLOWI
NG OPTIONS:-"
 230 PRINT
 240 PRINT " 1. LIST OF HOUSE NA
MES"," 2. SEARCH FOR A NAME"," 3
. DISPLAY STREET MAPS"," 4. ENTE
R/AMEND A NAME"," 5. START A NEW
 FILE"," 6. SAVE FILE ON TAPE","
 7. FINISH",," 8. SORT"
 241 PRINT
 250 PRINT TAB 6;"ENTER OPTION N
UMBER"
 260 INPUT Z$
 270 IF Z$<"1" OR Z$>"8" THEN GO
TO 260
 280 LET A=VAL Z$
 290 CLS
 300 GOTO A*1000
1000 PRINT TAB 8;"HOUSE NAME LIS
T"
1010 PRINT
1040 PRINT "YOU HAVE THE FOLLOWI
NG OPTIONS:-",,
1060 PRINT TAB 6;"1. A SELECTIVE
 LIST.",TAB 6;"2. A FULL LIST.",,
1070 PRINT TAB 3;"ENTER OPTION N
O.",,
1080 INPUT Z$
1090 IF Z$=" " THEN GOTO MENU
1100 IF Z$<"1" OR Z$>"2" THEN GO
TO 1080
1110 LET A=VAL Z$
1120 GOTO (A*200)+1000
1200 PRINT TAB 6;"ENTER INITIAL
LETTER"
1210 INPUT Z$
1220 IF Z$<"A" OR Z$>"Z" THEN GO
TO 1210
1223 CLS
1225 SLOW
1228 PRINT TAB 9;"  SEARCHING  "
1230 FOR F=1 TO N
1240 IF Z$=R$(F)(1) THEN PRINT T
AB 8;R$(F)
1250 NEXT F
1260 PRINT TAB 10;"END OF FILE",
TAB 5;"PRESS N/L TO CONTINUE"
1270 INPUT Z$
1280 FAST
1290 GOTO MENU
1400 CLS
1410 FOR G=1 TO N STEP 20
1420 FOR F=G TO G+19
1430 PRINT TAB 8;R$(F)
1440 NEXT F
1450 PRINT AT 21,6;"PRESS N/L TO
 CONTINUE"
1460 INPUT Z$
1470 CLS
1475 IF R$(F)( TO 2)="  " THEN G
OTO MENU
1480 NEXT G
1490 GOTO MENU
2000 PRINT AT 8,6;"NAME SEARCH R
OUTINE"
2010 PRINT " ENTER HOUSE NAME TO
 BE SEARCHED FOR.",," HOUSES IN
DIFFERENT AREAS MAY    HAVE IDENT
ICAL NAMES",,,"       ENTER AT LEAS
```

```
T 4 LETTERS","          PRESS N/L WH
EN READY."
2020 INPUT T$
2030 IF T$="" THEN GOTO MENU
2040 PRINT T$,"[illegible] SEARCHING [illegible]"
2050 PAUSE 50
2060 POKE 16437,255
2080 FOR E=1 TO B
2085 LET G=0
2090 FOR K=1 TO 20
2100 IF T$-N$(E,K)(1 TO LEN T$)
THEN GOTO 2300
2105 LET G=G+1
2110 NEXT K
2120 NEXT E
2130 CLS
2140 PRINT AT 10,8;"SORRY NOT FO
UND",,," DO YOU WANT TO ENTER AN
OTHER         NAME?";TAB 14;"Y/N"
2170 INPUT Z$
2180 IF Z$<>"Y" THEN GOTO MENU
2190 CLS
2200 GOTO 2000
2300 CLS
2305 LET D=1
2310 GOSUB 3010
2400 PRINT "PRESS [illegible] ""E"" END
, ""W"" WRONG  OR ""C"" TO COPY
"
2405 SLOW
2410 FOR H=1 TO 3
2420 PRINT AT G,15;"█"
2430 NEXT H
2450 FOR H=1 TO 3
2460 PRINT AT G,15;"<"
2470 NEXT H
2475 LET D=0
2480 IF INKEY$="" THEN GOTO 2410
2485 FAST
2490 IF INKEY$="W" THEN GOTO 210
5
2493 IF INKEY$="C" THEN COPY
2495 GOTO MENU
3000 REM PRINT MAP ROUTINE
3001 LET D=0
3002 LET E=1
3010 PRINT ;E;".";
3020 FOR F=1 TO 20
3030 PRINT TAB 6;M$(E,F);N$(E,F)
3040 NEXT F
3050 IF D=1 THEN RETURN
3060 PRINT "ANOTHER MAP? Y/N"
3070 INPUT Z$
3075 CLS
3080 IF Z$<>"Y" THEN GOTO MENU
3090 LET E=E+1
3095 IF E>10 THEN LET E=1
3100 GOTO 3010
4000 PRINT TAB 6;"AMEND MAPS ROU
TINE"
4010 PRINT
4020 PRINT "YOU MAY UPDATE AN EX
ISTING MAP, OR YOU CAN HAVE A BL
ANK SCREEN  TO DRAW A NEW MAP.",
TAB 5;"ONLY 10 ARE AVAILABLE"
4025 PRINT
4030 PRINT "YOU WILL BE GIVEN TH
E CHANCE TO RECTIFY ANY ERRORS L
ATER"
4032 PRINT
4033 PRINT "A SCREEN HAS ONLY 20
 LINES. YOU MAY NEED MORE THAN O
NE SCREEN."
4035 PRINT "NOTE A NEW MAP START
S LINE 1",,,
4040 PRINT "DO YOU WANT TO CONTI
NUE? Y/N"
4050 INPUT Z$
4060 IF Z$<>"Y" THEN GOTO MENU
4063 LET S=0
4065 CLS
4070 PRINT AT 21,0;"NEW OR EXIST
ING?  N/E.                      "
4080 INPUT Z$
4081 IF Z$<>"N" AND Z$<>"E" THEN
 GOTO 4030
4090 CLS
4100 IF Z$<>"E" THEN GOTO 5000
4130 PRINT "ENTER THE MAP REFERE
NCE NUMBER  IF NOT KNOWN PLEASE
ENTER ""0"""
4135 PRINT " YOU SHOULD FIND THE
 NUMBER ON    THE EXTREME LEFT OF
 THE MAP       SCREEN."
4136 PRINT AT 21,0;"ENTER NUMBER
."
4140 INPUT X$
4150 IF X$<"0" OR X$>"9" OR VAL
X$<>INT VAL X$ OR VAL X$>10 THEN
 GOTO 4140
4160 CLS
4170 IF X$="0" THEN GOTO 3000
4190 LET X=VAL X$
4200 PRINT
4201 PRINT X$;".";
4205 FOR G=1 TO 20 STEP 10
4210 FOR F=G TO G+9
4220 PRINT TAB 3;F;TAB 6;M$(X,F)
;N$(X,F)
4225 NEXT F
4230 GOSUB 5510
4235 CLS
4236 PRINT
4240 NEXT G
4270 GOTO MENU
5000 PRINT AT 9,7;"MAP ENTRY ROU
TINE"
5001 PRINT
5010 LET D=0
5020 PRINT B;".";TAB 3;C;
5050 SCROLL
5060 PRINT AT 21,0;"ENTER MAP-10
 CHARS, OR""ZZ""TO END"
5070 INPUT M$(B,C)
5080 IF M$(B,C)(1 TO 2)="ZZ" THE
N GOTO 5245
5090 PRINT AT 10,3;C;TAB 6;M$(B,
C)
5100 PRINT AT 21,0;"ENTER NAME -
 16 LETTERS               "
5110 INPUT N$(B,C)
5120 PRINT AT 10,16;N$(B,C)
5130 PRINT AT 21,0;"
                              "
5140 IF D=1 THEN RETURN
5150 LET N=N+1
5151 LET C=C+1
5152 LET J=1
5160 IF C=11 OR C=21 THEN GOSUB
5500
5170 IF C<>21 THEN GOTO 5050
5190 LET B=B+1
5191 LET C=1
5200 IF B<>11 THEN GOTO 5050
5220 PRINT "         THE FILE IS NO
W FULL"
5230 PRINT TAB 6;"PRESS N/L TO C
ONTINUE"
5240 INPUT Z$
5245 LET M$(B,C)="  "
5250 GOTO MENU
5500 REM CHECK ROUTINE
5505 LET X=B
5510 PRINT AT 21,0;"ARE THESE CO
RRECT? Y/N"
5520 INPUT Z$
5530 IF Z$="Y" THEN RETURN
5540 PRINT AT 21,0;"ENTER INCORR
ECT LINE NUMBER"
5550 INPUT P$
5552 IF P$<"1" OR P$>"9" OR VAL
P$<>INT VAL P$ OR VAL P$>20 THEN
 GOTO 5550
5555 LET P=VAL P$
5560 LET W=P
5565 IF W>10 THEN LET W=W-10
5570 PRINT AT 21,0;"ENTER MAP MA
X. 10 CHARACTERS."
5580 INPUT M$(X,P)
5585 PRINT AT W,6;M$(X,P);
```

```
5590 PRINT AT 21,0;"ENTER NAME M
AX. 16 LETTERS.        "
5600 INPUT N$(X,P)
5620 PRINT AT W,16;;N$(X,P)
5630 PRINT AT 21,0;"ARE THESE CO
RRECT?  Y/N                 "
5640 INPUT Z$
5645 PRINT AT 21,0;"
                          "
5650 IF Z$="N" THEN GOTO 5540
5660 LET J=1
5670 RETURN
6000 PRINT "              SAVE ROUTI
NE"
6020 PRINT AT 11,5;"START RECORD
ER",TAB 6;"PRESS N/L WHEN READY.
",TAB 8;"OR ""M"" FOR MENU"
6030 INPUT Z$
6035 IF Z$<>"  " THEN GOTO MENU
6040 SAVE "H/A"
6045 LET S=1
6050 GOTO MENU
7000 REM FINISH
7010 IF S=1 THEN GOTO 7050
7030 PRINT TAB 12;"WARNING","
THIS FILE HAS NOT BEEN SAVED","
 PRESS NO. 7 AGAIN TO FINISH"
```



```
7040 INPUT Z$
7050 IF Z$<>"7" THEN GOTO MENU
7060 CLS
7080 PRINT AT 10,10;">> FINISH <
<"
7090 GOTO 9999
8000 REM SORT ROUTINE
8020 IF J<>1 THEN GOTO MENU
8025 PRINT TAB 10;"SORT ROUTINE"
,"    THE SCREEN WILL BLANK OUT";
TAB 6;"*SO BE VERY PATIENT*"
8026 PRINT ,,"GO AND MAKE A CUP
OF TEA OR HAVE";TAB 9;"AN EARLY
BATH",TAB 7;"KEY N/L WHEN READY"
8028 INPUT Z$
8030 LET G=1
8040 FOR E=1 TO 6
8050 FOR F=1 TO 20
8060 LET R$(G)=N$(E,F)
8065 IF R$(G)(1)<"A" OR R$(G)(1)
>"Z " THEN LET R$(G)="   "
8070 LET G=G+1
8080 NEXT F
8090 NEXT E
8100 FOR K=1 TO G-1
8110 FOR L=K+1 TO N
8120 IF R$(L)>=R$(K) THEN GOTO 8
160
8130 LET T$=R$(L)
8140 LET R$(L)=R$(K)
8150 LET R$(K)=T$
8160 NEXT L
8170 NEXT K
8180 LET J=0
8190 GOTO MENU
9999 STOP
```

```
            **HOUSE NAME FILE**

YOU HAVE THE FOLLOWING OPTIONS:-

  1. LIST OF HOUSE NAMES
  2. SEARCH FOR A NAME
  3. DISPLAY STREET MAPS
  4. ENTER/AMEND A NAME
  5. START A NEW FILE
  6. SAVE FILE ON TAPE
  7. FINISH

     ENTER OPTION NUMBER
```

```
        AKENSIDE
        ALLYNBRAE
        ALNESS
        ASHCROFT
        ATTONBURN
        AULDEARN
        AVONDALE
        AYLWYN
        AYTONDALE
        BEECH HOLME
        BRAESIDE
        BRANDELHOW
        BRETTENLAW
        BRIERMEDE
        BURN BRAE
        CAIRNAVON
        CAMERTON
        CAMILLA
        COLDWELL LODGE
        DALEGARTH

     PRESS N/L TO CONTINUE
```

```
        >>SEARCHING<<
        WANCOURT
        WARDAINE
        WARENFORD
        WATENDLATH
        WEARHOLME
        WELWYN
        WERNETH
        WEST VIEW
        WESTLANDS
        WESTOE
        WILLOWBRAE
        WINGROVE
          END OF FILE
     PRESS N/L TO CONTINUE
```

```
                    SUN/LND RD VILS
 **          ********************
      *  H  *  ■ THE POPLARS
      *  I  *
      *  G  *  ■<AYTONDALE
      *  H  *  ■ FELLDALE
   D  *     *  ■ EAGLESFIELD
      *  H  *
   E  *  E  *  ■ CAMILLA
      *  W  *
   N  *  O  *  ■ GORSEWOOD
      *  R  *
   E  *  T  *  ■ BURN BRAE
      *  H        SCHOOL ENTRANCE
      *     *  ■ TUDOR LODGE
      *  L  *
      *  A  *  ■ ALNESS
      *  N  *  ■ DENESIDE
      *  E  *
      *     *  ■ GLENDEVON
 SS NNL   "E" END , "W" WRONG
 "C" TO COPY
```

Above are some example screen displays from the program. The top illustration shows the options available, then once you have entered the option you want you can list the house names or search for a specific name. The final example shows a street with the various houses with their names PRINTed alongside.

The prize of your dreams

Our reviewer, James Walsh, looks at the latest offering from Artic Computing and tries to win himself £10,000.



KRAKIT is a new program from Artic Computing which claims to be the ultimate adventure for the Spectrum/ZX81 with the added bonus of a generous prize if you should be the first to 'KRAKIT'. Retailing at £9.95 in this country, it certainly isn't cheap — but is it worth the cash? There is only one way to find out — cassette in, LOAD "RULES", press Play, sit back and wait.

Because of the complexity of the program, the rules and clues have been split up into two separate programs on the 16K ZX81 and ZX Spectrum. So, while we are waiting for the program to LOAD, I'll tell you something about the package itself.

The story so far...

The story is that your eccentric father has left you £10,000 in his will, but (and it is a big but) to claim the money, you must first solve 12 clues before you can gain access to the bank account in which the money has been placed. The prize money actually exists and just to make things completely fair, Artic Computing have released the program simultaneously in America to run on the Timex 1000. Artic are quite definitely going all the way to make it successful and fun — not only has there been a lot of press on the package, but Artic have set up a 24 hour telephone answer line so that contestants can keep themselves informed about the status of the contest and the amount of prize money now available (if the prize is not claimed the amount of money is increased on a weekly basis).

OK, it looks as though it has almost LOADed. Hurrying to get my registration form so that I can quickly claim my prize, I sit down waiting for the program to begin. You are first greeted with a menu which offers you five options: how to register, rules and play, see a sample clue, how to claim the prize and see the real clues.

All clued up?

The rules program is good. There is a nice sequence of graphics to start — a key moves across the screen and turns in a lock. After reading the rules, I pressed for option 3 and had a glance at the sample clue. After the solution has been explained to you, you are ready (?) to start on the real clues...which I did. So, LOADing the clues program, after a couple of moments (in ZX time, of course) up comes the title page:

WELCOME TO KRAKIT, THE ULTIMATE ADVENTURE

This certainly looks promising!

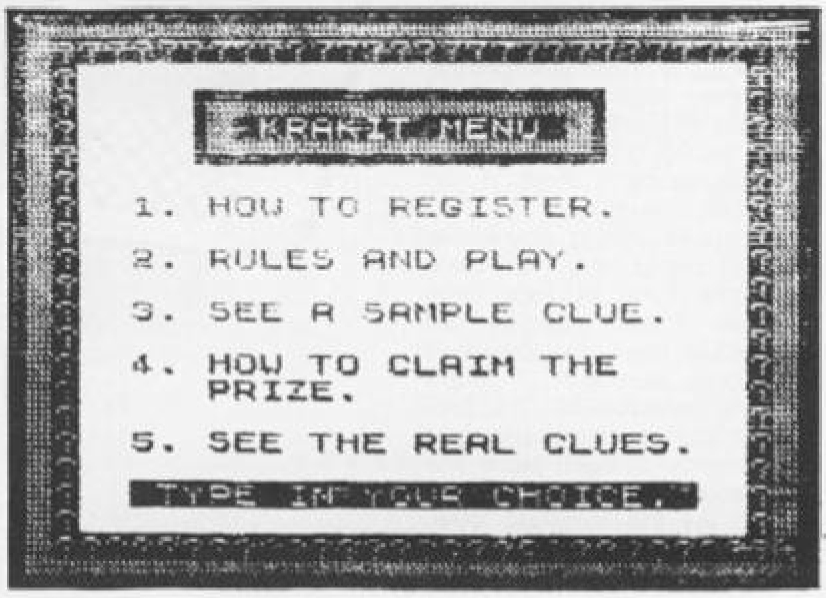
Then it's down to the real work of this program — the clues themselves. Their difficulty ranges from moderately difficult to downright impossible, but they are not all along the same lines as is so often the case with games.

Now for the one disappointment of the program — there are no graphics at all. Surely they could have 'jazzed' up the clues a little with some animated diagrams or moving pieces. That might sound a little silly, but it would be nice and would certainly brighten up the game. As it is the program has been written almost entirely in BASIC, which seems rather wasteful of space and time, though fortunately (as you will soon find out) the time factor is not so important!

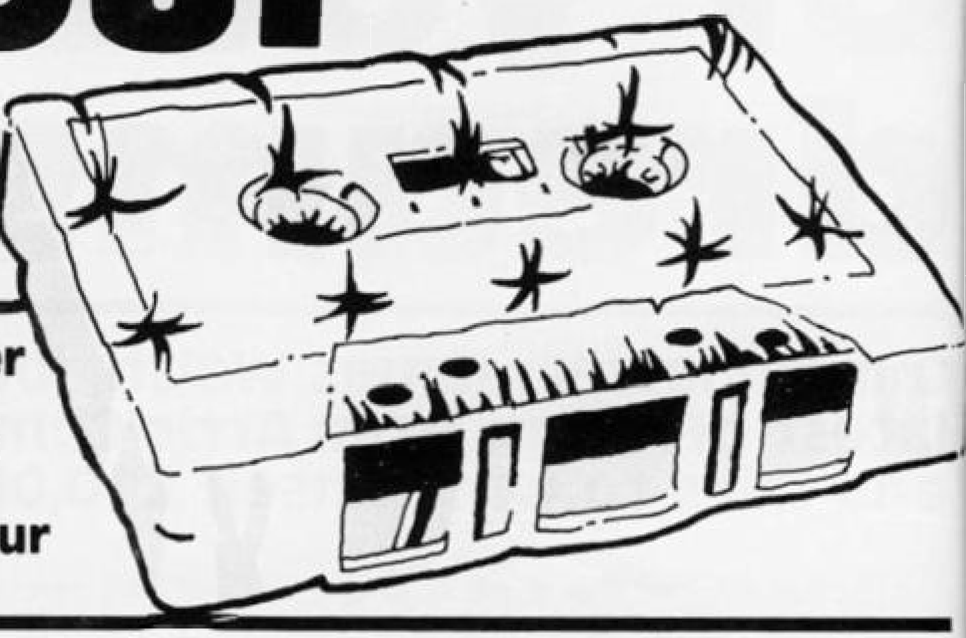
Actually playing the game is quite fun, but I fail to see the need to take the trouble to put on a cassette and load it all in and spend hours staring at a TV screen, when it could quite easily have been written out on a few pieces of A4 paper and sold for a tenth of the price.

As for the game itself, it is definitely good if you are into cryptic clues and mind-bending puzzles, but I would not say it was for the majority of games players. Needless to say, I have not filled in my registration form yet!

KRAKIT is priced at £9.95 and is available from Artic Computing Ltd, 396 James Reckill Avenue, Hull, North Humberside HU8 OJA.



Soft options for your ZX81



Puzzled by Dr. Nowotnik, after a glass of wine Nick Pearce settles down to checking out some of the software commercially available for your ZX81.

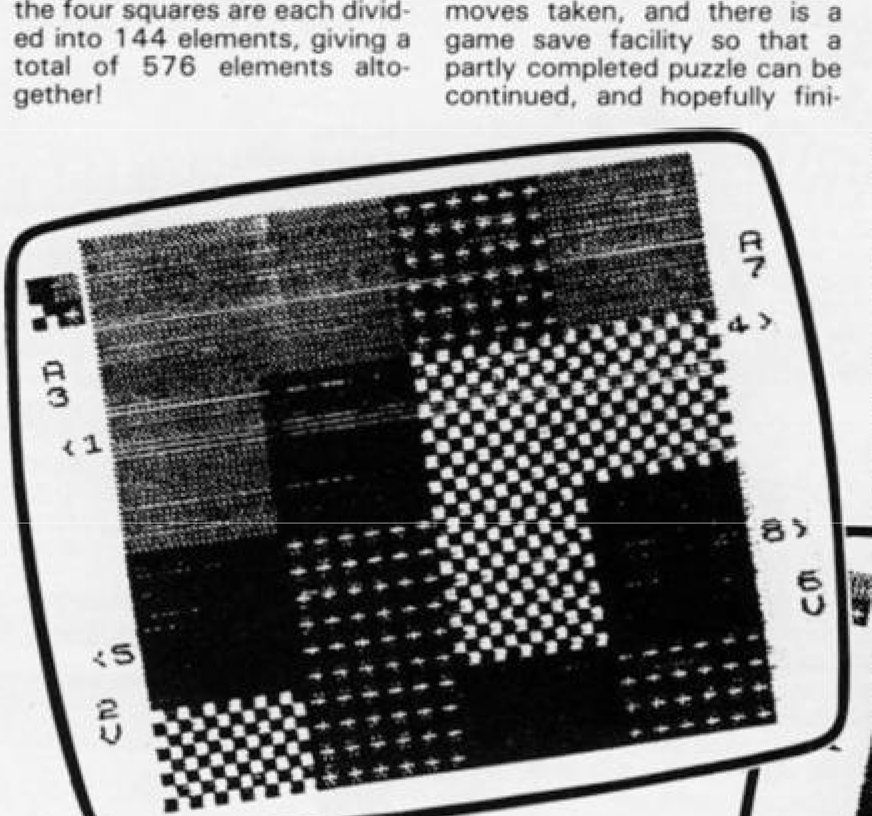
The Nowotnik puzzle and Other Diversions — Phipps Associates

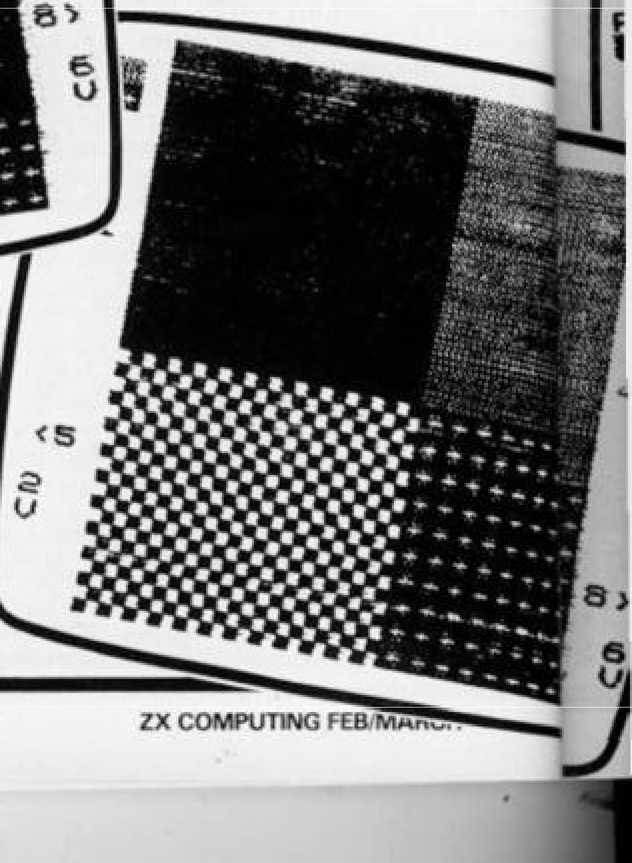
If you're bored with that dratted Rubik cube, then the Nowotnik puzzle is an original concept in computer games that might be just the thing for you. The chances are you'll be tearing out your hair in handfulls trying to solve the puzzle at the simplest level of play, knowing that there are four progressively harder levels to move onto if you ever do manage to get it worked out. "Easy — it's only two dimensional", I hear you say, as I myself commented — until, that is, I had seen the pieces shuffled and attempted a few exploratory moves. It is indeed a puzzle of merely two dimensions, but don't be under the misapprehension that it is, therefore, a doddle.

At the beginning, and when (and if!) completed, the puzzle is in the form of four large squares. These squares are shuffled by the computer using a random selection of eight possible movements. The problem is then to use these eight movements to get back the four squares in their original layout. At the simplest level, each of the four large squares is broken up into four by the shuffle, giving sixteen smaller squares; at the hardest level, the four squares are each divided into 144 elements, giving a total of 576 elements altogether!

If you complete the puzzle, you are told the number of moves taken, and there is a game save facility so that a partly completed puzzle can be continued, and hopefully finished, later. I would have liked an option to reduce the number of shuffles so that some logical method of solution could be developed by trial and error during the first few tries; the sight of a well-shufled puzzle on the first attempt is itself rather daunting.

The person who has mastered the magic cube will have a head start in tackling the Nowotnik puzzle. Conceptually, the problem is similar; it is relatively easy to complete one square, or face, but it is solving the whole puzzle without ruining those squares, or faces, already completed that is so infuriatingly difficult.





A very addictive game, and as is so often the case with this kind of competition of self versus the computer, it is very hard to admit defeat and one tends to persist, thinking that a few more moves will have it cracked. Perhaps it is fortunate that the ZX81 is less portable than the cube, otherwise it would undoubtedly make its appearance at parties with the nonchalant admission, by devotees, of the number of moves or time taken to solve the puzzle.

A dynamic duo

Two other games, routine in comparison to the puzzle but nonetheless absorbing, complete this cassette and make it good value for money.

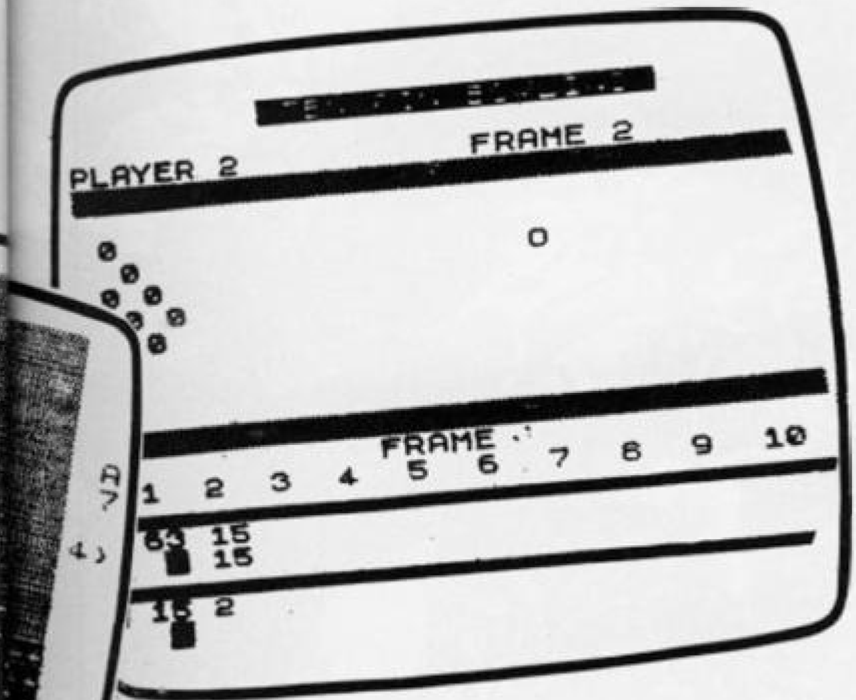
In Demolition, a wall appears at the bottom of the screen and the object is to knock down as many bricks as possible using a '*' fired repeatedly at the wall. There is a catch — the wall moves upwards at a slightly faster rate than you can knock bricks out, and also more walls appear; eventually the game ends when a wall reaches the top of the screen. Points are awarded for each brick removed, and the total for the game is displayed so that a competition between a number of players can be held. A fast moving interactive game and completely idiot proof — my four year old daughter played for about an hour without managing to crash the program!

Finally, Tenpin simulates a standard game of tenpin bowling. The alley is displayed at the top of the screen, with score cards for either one or two players displayed below. The game is played over ten frames. When bowling, you indicate the strength of delivery; a stronger delivery is less accurate but increases the number of pins which fall if you make a hit. Not being a tenpin bowling afficionado, I was intrigued by the scoring which apparently is true to life and is coupled with 'strikes' and 'spares'. The graphics are fair; the ball disappears when it hits the pins and the effect as the pins fall and also disappear is rather disconcerting. Nonetheless, a novel simulation to complete the package.

All the games on this cassette require a ZX81 with more than 1K, and each program is duplicated on the reverse side. The programs can all be LISTed, and instructions for making security copies are given. Dr. David Nowotnik, the author of the three programs, is a member of the Aylesbury Computer Club.

'The Nowotnik puzzle and Other Diversions' costs £5.00 and is available from Phipps Associates, Mail Order Dept., 99 East Street, Epsom, Surrey.





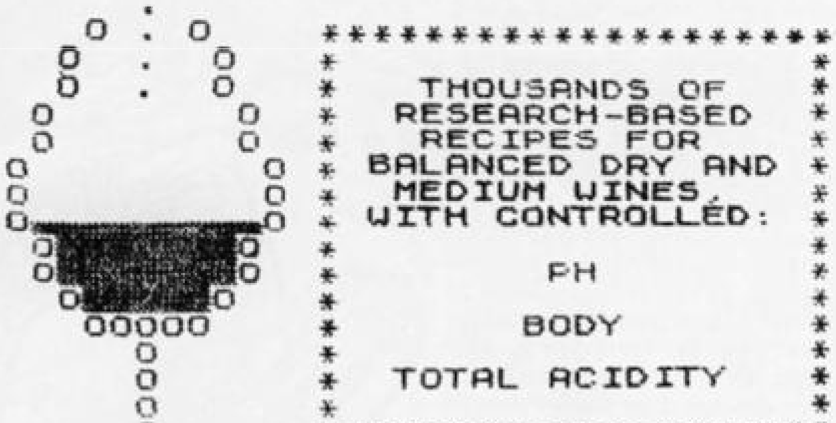
Computerwine — Computerwine

If you make your own fruit wines, you might find this package of interest. It is aimed primarily at the more experienced winemaker, and is a tool to enable precisely balanced wine recipes to be designed quickly and accurately. For the amateur, it shows that there can be more to making wine than religiously following recipe books — perhaps leading to the development of winemaking skills.

The package is based on the results of recent and apparently extensive research which has led to the discovery of some of the scientific principles which govern the balance of wines. It is now possible to select through these principles the ingredients of the must to give a wine of the required acidity, body and alcohol content. Flavour is not scientifically defined but is left up to the winemaker through the choice of a large selection/combination of fruits — although hints for achieving a wine of good flavour are given. The computer programs cover the design of balanced wine recipes for dry and medium wines (Computawine intend to make programs available for sweet and dessert wines soon).

Instructions come in a 22 page manual, which is well written and extremely helpful — a pleasant change, I feel; so often, otherwise good computer software is let down by the poor quality of the supporting documentation. As well as describing the background to the experimental work, together with the theory behind a 'balanced' wine and giving a comprehensive explanation of the program, it also gives some useful practical hints on successful LOADing and the merits of various cassette recorders and makes of cassette. It gives a lot of useful information of a more general nature to help the winemaker, although probably not sufficient to instruct a complete beginner in the art of winemaking.

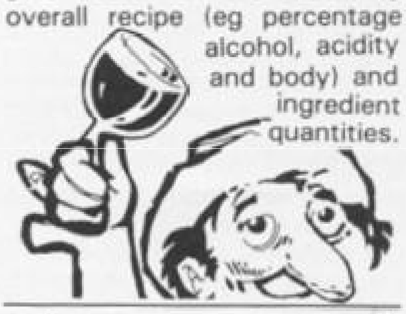
Five separate programs are recorded on the cassette, and each is duplicated on the reverse side. Firstly, there is the Preface; this takes three minutes to LOAD and runs automatically, starting with some nice graphics. It then displays a summary of a few of the more important points given in the manual. A nice touch (although I feel it is largely unnecessary) is that the manual itself is sufficiently comprehensive and gives full instructions as well as examples.



A wine time

The other four programs contain the calculations for the wine balances. The selection of the balance, and hence which of these four balance programs to use, depends on the wine one wishes to make. Balance 4 should give a quick, dry table wine; balance 3, a wine which is a little improved in quality, fuller in body and slightly sharper, but taking longer to make; balance 2 and 1 give wines of fuller body and more flavour. Each of these programs takes about six minutes to load.

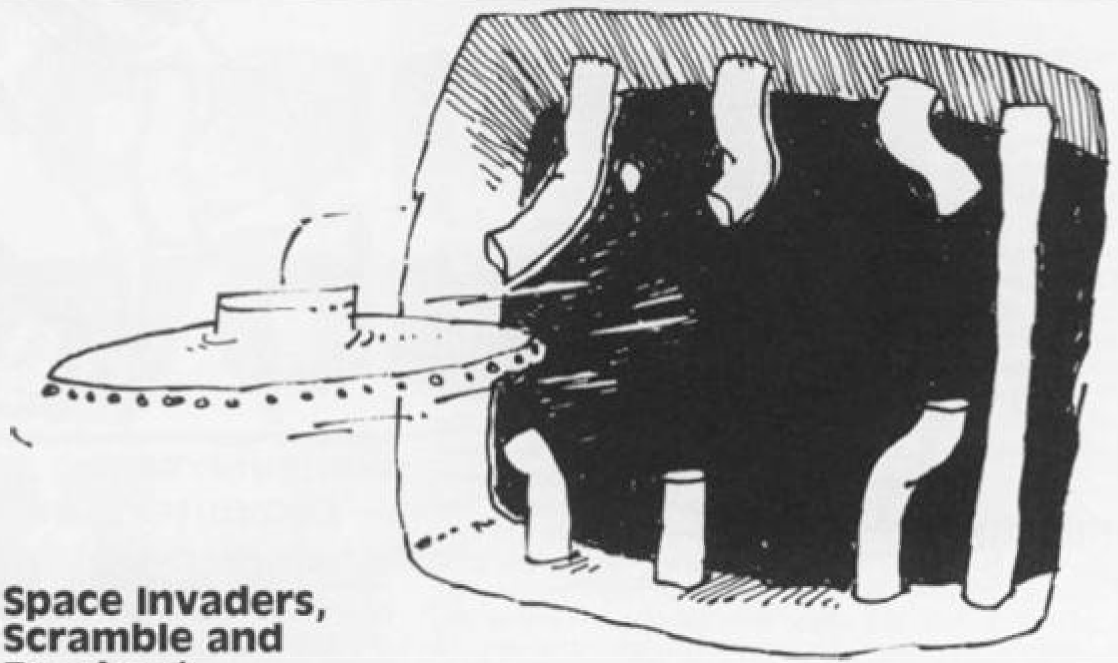
These programs are used to create wine recipes by specifying (from a list given in the manual) which of the two-ingredient sub-recipes are to be used (up to eleven can be combined), and the proportion of each. The programs go on to give data on the resulting overall recipe (eg percentage alcohol, acidity and body) and ingredient quantities.



In summary, a novel application for the ZX81 and a very well-produced package in all respects. The programs are easy to use, my only criticism being the time taken to LOAD which might tend to put off the casual winemaker; and one really does need a cassette recorder with a digitiser, or fast forward/cue facility, to be able to locate quickly the desired balance program. One could overcome this, of course, by SAVEing each balance on a separate cassette — instructions for SAVEing the programs are given.

I cannot comment on the quality of wines made using recipes developed from the Computawine programs — you'll have to wait a year or so, I'm afraid, to find out the success of my efforts! Both wine making and computing are time consuming hobbies, and one tends to preclude the other — too much wine and the humble ZX81 keyboard starts to resemble the cockpit display of Concorde in complexity. For those who can combine both, this package may be well worth looking at.

'Computawine' costs £7.95, and is available only from Computawine, 9 Laburnum Way, Etwall, Derby.



Space Invaders, Scramble and Breakout — Microgen

Space Invaders is Microgen's version of the arcade type game of the same name in which you defend the Earth from successive armies of invaders. On LOADing the program, you are presented with a choice of three speeds (normal, fast and superfast). The invaders are displayed at the start of the game and there is a short pause before the action starts to allow you to prepare yourself. You have three lives (bases), and lose one each time a bomb from the invader fleet lands on you, or if you allow any invader ship to get down to the level of your shields. The technique is to knock out the lower ships first with your laser beam.

Ten points are awarded for each invader ship hit, and if you clear the first wave, another will take its place, and another . . . Each time this occurs the game speed is increased slightly. The game performed well and I found it enjoyable without being positively addictive. However, I thought the action was somewhat jerky, particularly at the slowest speed. I was told by an experienced player who had a go that it bears the most similarity to the arcade game when it is played in the superfast mode — too fast for me!

The program includes the option to use Microgen's joysticks, which I think would make a significant improvement over the use of the touch sensitive keyboard for interactive games of this sort. A nice finishing touch is the display at the end of each game giving the score for that game and the highest score so far, thus enabling competitions to be played.

This game is also marketed by Sinclair under the Psion label with Bomber, also by Microgen, on the reverse side.

Scramble is another fast interactive arcade type game. In this one, you are flying low over mountainous enemy terrain, the object being to keep your ship in the air for as long as possible. At the same time, you have to shoot down as many of the alien attackers as you can,



and there are air-to-air and ground based missiles to contend with as well. Points are awarded for each alien and air-to-air or ground missile you hit.

Your fleet is small — three ships — and only one is in the air at a time, although you can get a bonus ship if your score reaches five thousand points. Fortunately, you carry an unlimited supply of ammunition plus one 'smart' bomb in each ship which will destroy everything in sight. The nearer you get to the enemy base, the more frenetic become the alien attacks on your ship.

Again, joysticks can be used and would be a definite advantage with this game. It is particularly difficult with the touch sensitive keyboard of the ZX81 to both maintain and achieve forward and up and down movement as well as battling with the aliens. As with Space Invaders, the most recent game score and 'highest so far' score are displayed to facilitate competitions.

Another brick...

Finally, from Microgen, Breakout is the traditional 'ping pong' video game. Again you have a choice of three speeds (normal, fast and superfast) and three lives per game. You score ten points for each brick removed from the wall, and after each hit, the speed of the ball increases slightly. There are seven angles of rebound off the bat, and one life is lost each time the ball misses the bat. You win one thousand points for completely clearing the wall (not much chance of that with yours truly at the controls, but with practice and at the slowest speed, who knows...), and a record of the highest score is kept. A well-written and enjoyable version on this not terribly inspiring theme.

All these three Microgen games require more than 1K and are written in machine code. They are 'locked', ie the programs cannot be LISTed and, for example, modified by the user. They perform well, are enjoyable, and are idiot and crash proof. However, with only one game on each cassette, they are in my opinion rather overpriced. There is only one recording on each cassette, although the review copies all LOADed first time.

'Space Invaders', 'Scramble' and 'Breakout' cost £3.95 each and are available from Microgen, 24 Agar Crescent, Bracknell, Berkshire.

Gamestape 6, Breakout — J. K. Greye

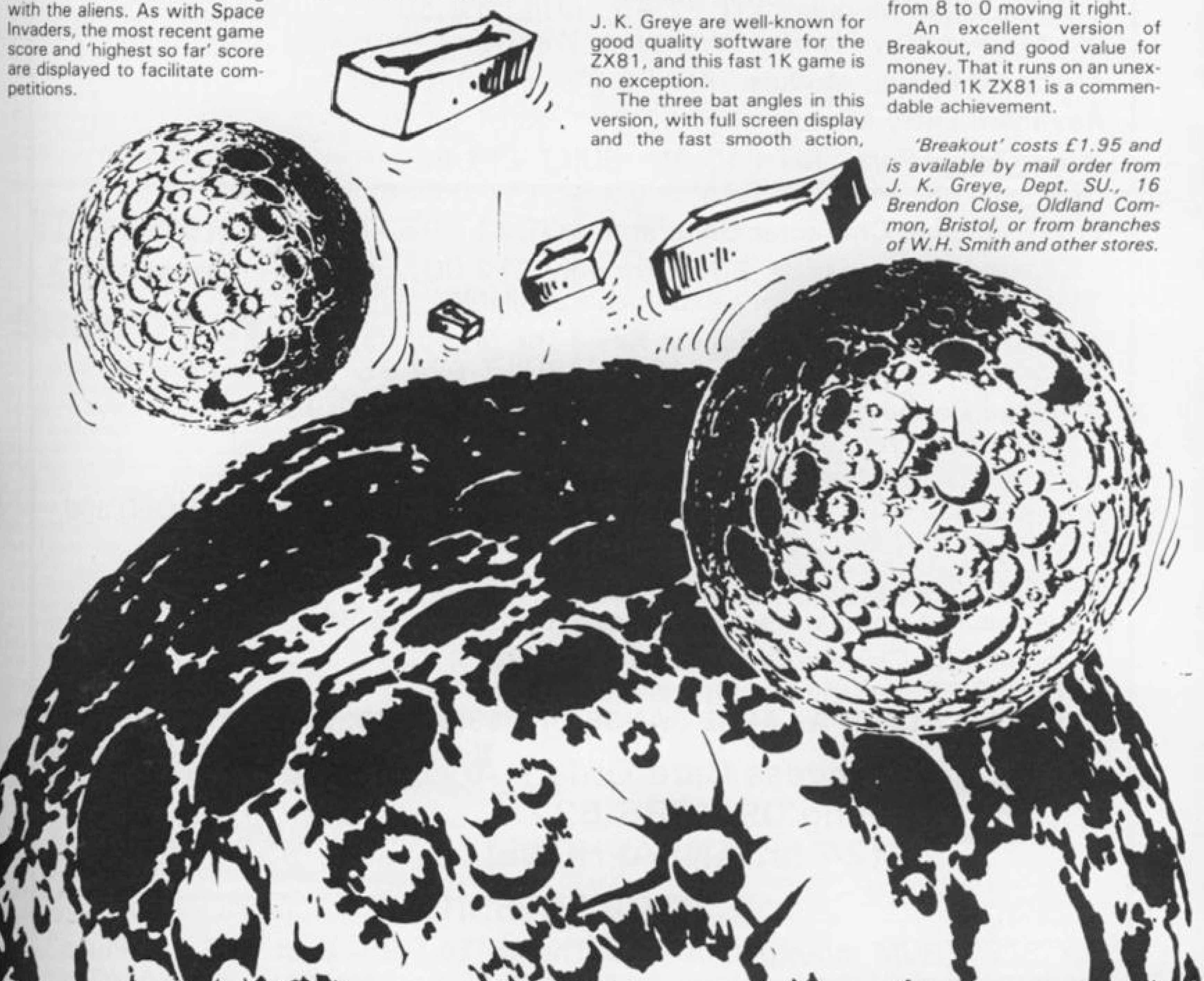
J. K. Greye are well-known for good quality software for the ZX81, and this fast 1K game is no exception.

The three bat angles in this version, with full screen display and the fast smooth action, make it a treat to play. It is written in machine code, and the speed (choice of fast, medium and slow), and the bat size (large or small) can be varied using POKE commands. Although written in 1K, this game can also be played on a ZX81 with 16K RAM pack (by POKEing RAMTOP *before* LOADing the program — full instructions are given).

Nothing as straightforward as the usual Breakout wall of bricks on this one either. The wall is built up of pound signs which convert to dollars when hit. Ten balls are available and a maximum score of $240 can be achieved (but not by me!). The instructions state that bat movement is effected by using the cursor keys 5 to 8; on the review copy, however, any key in the top row from 1 to 5 moved the bat to the left, keys from 8 to 0 moving it right.

An excellent version of Breakout, and good value for money. That it runs on an unexpanded 1K ZX81 is a commendable achievement.

'Breakout' costs £1.95 and is available by mail order from J. K. Greye, Dept. SU., 16 Brendon Close, Oldland Common, Bristol, or from branches of W.H. Smith and other stores.



HAVEN HARDWARE
ZX PRODUCTS & JUPITER ACE

SAE for details. 4 Asby Road, Asby, Workington, Cumbria CA14 4RR.
Prices include VAT and P & P.
Some of the products are also available from the following agents:
THE BUFFER SHOP, 374A STREATHAM HIGH ROAD, LONDON SW16
MICROWARE, 5 ST PETER'S LANE, LEICESTER.
PHILLIP COPLEY HI-FI, 7 CLIFFARD COURT, OSSETT, WEST YORKSHIRE.

TRADE ENQUIRIES WELCOME

ZX80 & ZX81 HARDWARE

Repeating Key Module KIT £3.95 BUILT £5.95
(As reviewed by *Popular Computing Weekly*)

Inverse Video Module KIT £2.95 BUILT £4.50
(As reviewed by *Popular Computing Weekly*)

Keyboard Beeper Module KIT £6.95 BUILT £8.50

Keyboard Entry Module KIT £2.95 BUILT £4.95

Input/Output Port KIT £10.95 BUILT £14.95
(Can be used with 16K RAM and printer without motherboard)

Programmable Character Generator KIT £17.95 BUILT £24.95

3K Memory Expansion BUILT for only £12.00
The most reliable memory expansion available.

Full-Size Keyboard with Repeat Key—The first of its kind
Built version plugs in. KIT £19.95 BUILT £24.95

ZX Edge Connector 23-Way Long gold plated pins £2.95
ZX Spectrum version £3.50

ZX SPECTRUM SOFTWARE CASSETTES

Fruit Machine Colour graphic representations of fruits. Including HOLD and GAMBLE. £4.95

Solitaire Alpha-Numeric move entry. £4.95

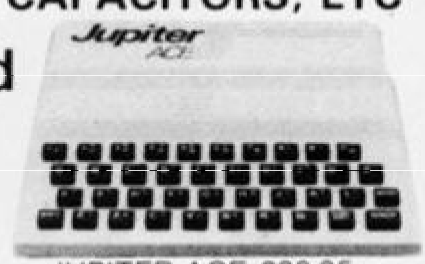
Patience The popular card game (cheat proof). £5.95

Mancala The Chinese Logic Game. £5.95

REPULSER Repel the alien invasion. £4.95

WE ALSO STOCK A LARGE RANGE OF RESISTORS, CAPACITORS, ETC

Access Card Orders Accepted
Ring 094 686 627
(24 hr. ANSAPHONE)



JUPITER ACE £89.95

STOP PRESS!!

ZX SPECTRUM input/output port. The kit £11.95. Built £15.95. Plugs onto the back of the spectrum doesn't require motherboard. (uses *BASIC* in and out commands).

What's in store for '83?

At a recent press conference, Nigel Searle of Sinclair Research was asked many of the questions which have been on everyone's lips over the last few months.

The Microdrives were, of course, one of the first topics to be brought up. Mr Searle announced that he expected to see the Microdrives released in the first quarter of 1983. As to why there was a delay in their introduction, journalists were told that the design of the Microdrive was altered at a late date to incorporate some improvements. Mr Searle explained why no details had been given on the Microdrive, "There is just no reason to give them (our competitors) three or four months lead time. We expect the Microdrives to improve the sales of the Spectrum, but I would expect the Spectrum to sell on its own anyway." Mr Searle went to explain that the Microdrives will be using "a totally new material for the recording medium. It's not been used before in computers."

Talking about Sinclair Research's sales abroad, Mr Searle had this to say about their growth in the Japanese computer market. "We've not made anything like the impact on the Japanese market we would have hoped, but we will be tackling it. The realistic alternative is to do it on one's own. It's a difficult job to do, but it's a challenge. If we took that approach, we would open an office in Japan, but as a matter of choice we would not manufacture in Japan."

Moving onto the subject of the competition, which seems to have made itself more obvious over the last few months, Nigel Searle had some fairly cryptic comments to make. When asked if the price of any of their products would be coming down due to the introduction of products such as the Oric, he replied "If they knew what Sinclair were going to do, they might well have cut their prices." And on Binatone, Nigel Searle really set the cat among the pigeons with these comments — "I think we will worry about that (the Binatone computer) when they release the machine. If we really believed the rumours we've heard, we would have released the ZX83 two months ago."

Nigel Searle shouldn't have been surprised at the avalanche of questions resulting from his throwaway comment. Hedging a little on the 'ZX83', Nigel Searle had this to say, "We are going to try to produce a machine which is quite different from any other on the market. We want to create a new segment of the market — it's not a matter of moving up or down the personal computer market."

The High Street Spectrum

Just as the ZX81 began life as a mail order item only to end up as the darling of the chain stores, so to has the ZX Spectrum found its way onto the High Street shelves.

To an astonishingly quiet fanfare, WH Smiths began stocking the ZX Spectrum in late November, 1982. And there was good reason for their silence — the first batch of Spectrums delivered to 66 of the larger WH Smiths stores were sold out within two to three hours! However, now stocks of the Spectrum are widely available, WH Smiths are proudly beating their chests over their computer coup.

The 16K Spectrum will sell at £125 and the 48K version will be £175, both prices inclusive of VAT. An extensive range of compatible software, including WH Smiths' own-brand range, is now available in the larger branches together with a new WH Smith Computer Carry Console at £14.95. A comprehensive range of computer books and magazines is also available in the stores' 'Computer Know-How' departments.

You should be able to purchase a ZX Spectrum, and accompanying software and publications, from any of the larger WH Sm'ths stores in England.



Better read...

Hilda Bernstein, the South African womens' and black rights' campaigner, has won the first annual £5,000 Sinclair Prize for Fiction for her novel, Death is Part of the Process.

Due to the very high quality of the entries, the sponsors agreed, at the judges' recommendation, to provide for this year only, a second prize worth £2,000 and two third prizes worth £500 each. Announcing the awards at an evening reception in London, Clive Sinclair also presented the second prize to Gill Edmonds for her novel, The Common, and third prizes to Aviot John for Chasing Cursors and to Philip Latham for Sara Singing. The judges for the Sinclair Prize for Fiction were Frank Kermode, David Caute, Richard Hoggart, Mervyn Jones and Polly Toynbee.

Explained Clive Sinclair as he was about to award the prizes, "We are delighted to make these three additional awards in the Prize's inaugural year, because the authors responded so well to our purpose of encouraging new writing on critical issues. In particular, Hilda Bernstein's winning novel completely fulfilled the Prize's criteria for high literary merit combined with contemporary social and political relevance."

To be published in the New Year by Sinclair Browne, Hilda Bernstein's novel, Death is Part of the Process, is based on Ms Bernstein's own experience of political conflict in South Africa. An inside view of people and organisations that plan and carry out acts of sabotage, it deals with the tensions of actual events and the complex moral dilemmas they provoke.

The National Book League are now taking entries for the 1983 award. If you fancy trying your hand at writing a novel for the Sinclair Prize for Fiction, write for further details from Barbara Buckley, National Book League, Book House, 45 East Hill, Wandsworth, London SW18 2QZ.

Spectrum for the disabled





As a result of close collaboration between Possum Controls and Sinclair Research, the ZX Spectrum is now available in a form which is easily operated by physically disabled people.

The Possum ZX Spectrum comes in three versions: the Desk-top Scanning device, for constant use in the office or school; the Brief-case Scanning model, a cased version allowing accessories to be carried in the lid; and the Expanded Keyboard model, for people with gross movement or tremor.

The scanning models are operated by using any Possum input to scan the light around the front panel which is a complete replica of the Spectrum keyboard. In addition, an eight-way method of selection can be employed using a joystick or footskate to provide faster selection.

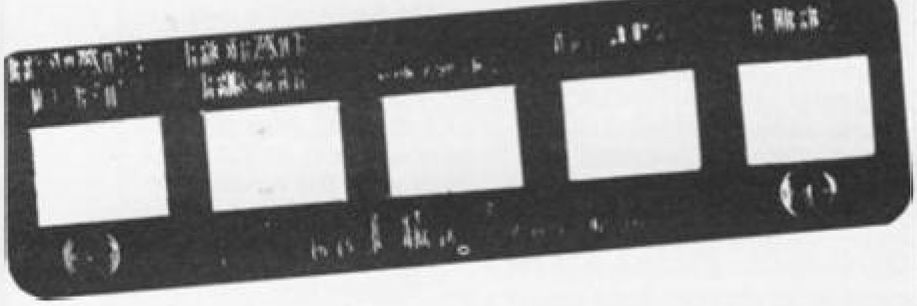
The Expanded keyboard version is an enlarged keyboard replica of the Spectrum keyboard and has the keys recessed and spread out to facilitate operation. There are also speed, delay and tone controls which can be adjusted to suit individual operation requirements.

As well as the regular Sinclair manuals, there is a Possum computer handbook which details the setting-up and operation of the Possum machines. All systems are available with 48K RAM. Possum Controls will also be making recommended educational software available.

The Expanded keyboard model can also be used in conjunction with the 16K ZX81 and, as with the other Possum products, the devices are compatible with the Sinclair range of accessories.

The price of the units is very much dependent on the input devices used with the machines and for further details, you should get in touch with Possum Controls Ltd, Middlegreen Road, Langley, Berkshire SL3 6DF or 'phone 0753 79234.

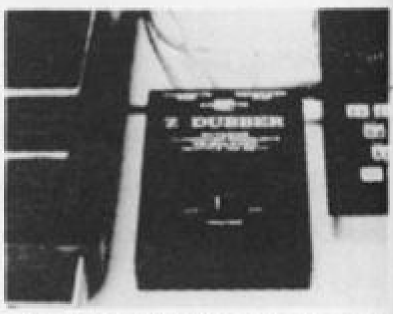
The keys to success?



'Why hadn't it been done before' seems to be the main question being asked about this product. It's so simple and yet so useful.

Quicksilva have announced keyboard overlays which will be provided free with their games Meteor Storm and Space Intruders. The overlays fit neatly over the required keys of the Spectrum for each particular game. Then, as Quicksilva say themselves, there will be 'no more fumbling with keys and no more thrusting when you should be firing!'

Loading problems?



If you experience any problems loading programs on cassette into your ZX81, you may wish to consider an American product that has just come on the market.

Called the Z-Dubber, the unit (as you can see in the photograph) interfaces between the ZX81 and the tape recorder boosting the reproduction of the sound so that even the most difficult cassette program will load easily. Additionally, the Z-Dubber allows you to connect two cassette recorders together to create good backup copies of your programs.

Battery-operated, the unit sells for $29.95 in the United States, although you will probably have to add a small shipping fee if you order one from the UK. For further information on this device, get in touch with Bytesize Computer Products, PO Box 21123, Seattle, WA 98111, U.S.A.

Measure for measure



Two new products are available to enhance the operation of the Sinclair range of computers.

The first is the Measurement Module, which is designed to operate with a ZX81, offering science teachers access to a versatile, data gathering system with a wide variety of measurement, data processing and display options.

The Measurement Module comes complete with a variety of programs which operate the Module and select the appropriate physical variable to be measured (eg voltage, current, magnetic field or temperature). Although there are no switches on the Measurement Module, it is a multi-range, multi-channel instrument. All range and channel selections are performed by the program. All the Module's functions are controlled via single key strokes from a main menu; no knowledge of programming or electronics are needed to operate the instrument.

In addition, the instrument offers outstanding protection to the user's computer against the inadvertent application of high voltage (all digital inputs/outputs to the computer pass through optical isolators, giving up to 1.5KV overload protection). The inputs to the Module itself are also capable of withstanding a very high overload thus making it suitable for use by the students themselves.

Software is the key to the instrument's flexibility. By measuring voltage and current, the program can multiply these quantities and obtain the power value in Watts. In addition, by integrating over a period of time, the instrument can act as a digital Joule meter. Data can be gathered and displayed in large digits for all the class to see or in columns of data taken at regular intervals, or automatically plotted on a graph as the experiment continues. But perhaps the unit's greatest strength lies in its ability to measure two variables simultaneously and then plot one variable against the other, thus enabling a large number of relationships between physical quantities to be rapidly investigated.

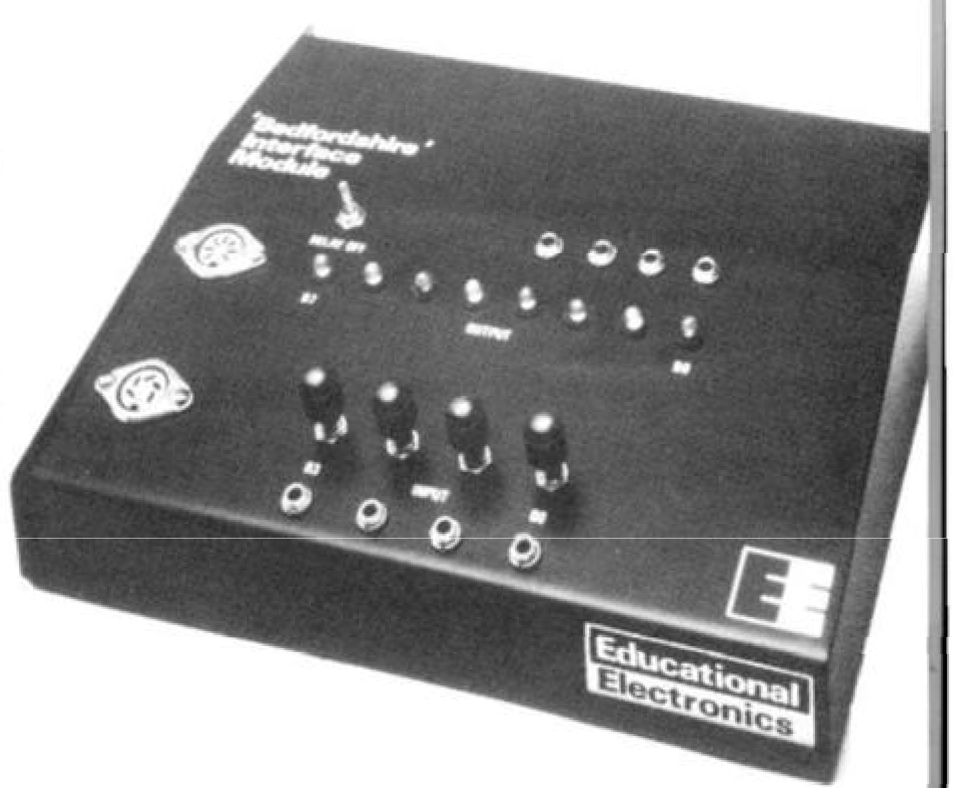
Whilst the Module is primarily aimed at physicists, it will also accept inputs from a variety of standard biological probes found in educational establishments.

The second unit available is the Interface Module which is designed to enable a Spectrum or ZX81 to be used to control robotic arms, solenoids, hydraulic and pneumatic valves, and stepper motors, etc.

The Interface Module contains four (optionally six) relays capable of switching 5A each and a stepper motor driver IC. In addition, it has a built-in traffic light sequence and all inputs to the computer pass through opto-isolators so that the computer is not damaged by very high voltages accidentally applied to the input sockets. A range of accessories such as a joystick switch, speaker box, A to D and D to A converter also plug into the Module.

Although originally designed for use in technology education by the Advisory Unit of Bedfordshire's Education Authority, the Interface Module could well be of use to the enthusiastic robotics user.

The Measurement Module and the Interface Module are priced at £98.00 and £79.50 respectively. For further information on the devices contact Educational Electronics, 30 Lake Street, Leighton Buzzard, Bedfordshire LU7 8RX or 'phone 0525 373666.



That's handy

Midwich Computer Company Ltd have announced high quality analogue joysticks for the ZX81 and Spectrum.

The joysticks are made of injection moulded plastic and are designed to sit comfortably in the hand. Incorporated into the joysticks are potentiometers which are claimed to have a life expectancy in excess of 200,000 operations (just think how many alien invaders you could kill in that time!).

A push button is built into the handle which can be operated, for example, as a 'fire' button. Each joystick or pair of joysticks are fitted with DIN plugs to suit the Sinclair range of computers. Also, as the ZX81 and Spectrum do not have an A/D converter built in, Midwich have also designed a low cost, high speed, four channel joystick controller board. This plugs into the expansion slot of the relevant machine, and incorporates an edge connector so that other peripherals can also be added.

The joysticks are available at £15.98 per pair from Midwich Computer Company Ltd, Rickinghall House, Hinderclay Road, Rickinghall, Suffolk IP22 1HH. Telephone enquiries can be made on 0379 898751.

Sinclair software



Sinclair Research have announced a further collection of software cassettes to complement their existing range of business, educational and games packages.

Priced from £2.95 to £14.95, there are 18 cassettes available in the new range and have been specially written by Melbourne House, Games of Skill, Artic, Psion and ICL.

There are four adventures for the 48K Spectrum including Planet of Death, Inca Curse, Ship of Doom and Espionage Island (Planet of Death will also run on the 16K Spectrum). Two other cassettes for the 48K Spectrum are Collector's Pack, which allows you to keep up to 1,500 records of stamps, LPs, etc; and Club Record Controller, which enables clubs to hold records of up to 300 members including their names, addresses, etc. A Reversi game is also available for the 16K Spectrum.

For the ZX81, Sinclair have made available 10 new software cassettes. For the 16K ZX81, there are versions of the adventure games Planet of Death, Inca Curse, Ship of Doom and Espionage, as well as Reversi, and two cassettes with Thro' the Wall and Scramble, and Super Glooper and Frogs. For the 1K ZX81, there are two packages: one providing a game of chess and the other, a collection of 11 games such as Slot Machines, Slalom, Space Pirate and Maze.

A useful ZX81 Toolkit has also been announced for the 16K ZX81 providing nine new functions: RENUMBER, DELETE, MEM, DUMP, FIND, REPLACE, SAVE and APPEND, and REMKILL.

The Hobbit habit



Completing the new catalogue of Sinclair software is their piece de resistance, The Hobbit. Based on the fantasy land created by JRR Tolkein, the player takes on the role of Bilbo, the hobbit. This adventure program is presented in words and full colour graphics and is designed to run on the 48K Spectrum.

Perhaps the most unique factor of this program is that the user instructs the computer in completely ordinary English sentences. The Hobbit program is capable of very sophisticated communications using adjectives, adverbs and multiple sentences which the Spectrum understands. In all, the program has a built-in library of 500 words.

You get to meet all of the characters from Tolkein's book, including Gandalf, Thorin, Gollum, and many others, and best of all, they not only interact with you but act independently as well. Due to this feature, each time you take part in the adventure, events will proceed in a slightly different way, and the further you get into the land of Tolkein's Hobbit, the more different it will get.

Written by Melbourne House, The Hobbit comes complete with a copy of Tolkein's book and instructions to play. Priced at £14.95, for information on this or any of the rest of the Sinclair range of software, contact Sinclair Research Ltd, 6 King's Parade, Cambridge CB2 1SN or 'phone 0223 353204.

May the force be with you

The people up at Work Force must have been busy of late judging by the huge selection of software they have dreamed up for the Spectrum and ZX81.

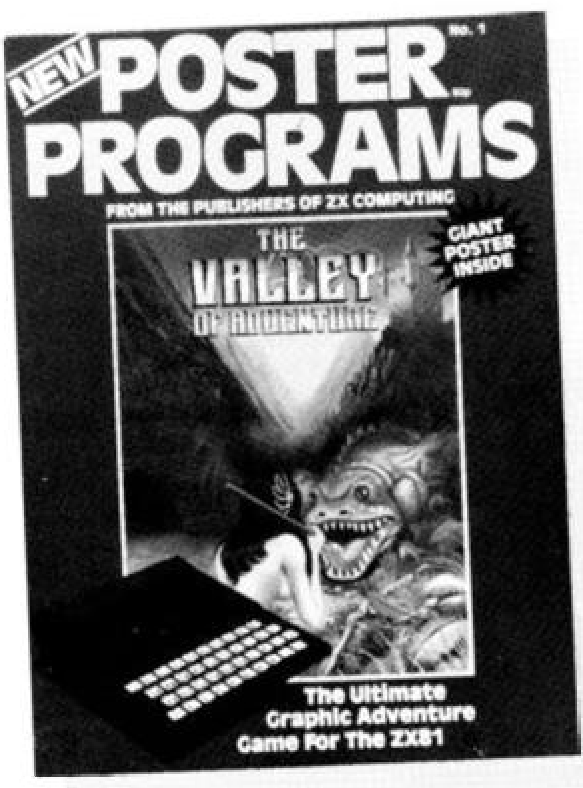
Looking first to the Spectrum software, they have devised a package called Programmer's Dream which provides the user with a number of extra features such as RENUMBER, BLOCK MOVE, BLOCK ERASE, LINE ERASE, SEARCH and REPLACE, DUMP VARIABLE/STRING, as well as commands to make a quick check on the program and variable size. The features are called from just one line of BASIC and operate instantly. The Programmer's Dream is priced at £7.00.

Other software for the Spectrum includes a Disassembler in 1,500 bytes; Di-Loader, a machine code loader to use with a disassembler; and Renumber Delete, a fast machine code loader which allows the user to define where to renumber from, what the new start should be and specify the increment. The Disassembler, Di-Loader and Renumber Delete are priced at £5.50, £7.00 and £5.00 respectively.

And if you've got a 16K ZX81, Work Force haven't forgotten you. Among the programs available are Adventure in Time, an extravaganza based in Australia where you have to use your time machine to travel forwards and backwards in time to save mankind from nuclear destruction; and A.D.V.E.N.T., an eight part program, the function of which is to create adventure games (and if that sounds confusing, a mini adventure, Treasure Mountain, is included to show you how it's done). These two packages are priced at £6.95 and £5.95 respectively. Other programs for the 16K ZX81 include Address, which allows the user to set up files of addresses on tape; ZX80 Convert, which will allow you to run ZX80 programs on your ZX81; and Greatest Games 1, a package which contains 10 games such as Pac Maze, Gobler, Star Defender, Vampires, Minefield and others. These last three titles are priced at £4.95, £4.95 and £5.00 respectively.

For more information on all these titles and the rest of the Work Force range of software, contact Work Force, 140 Wilsden Avenue, Luton, Bedfordshire or 'phone 0582 454456.

Poster problems?



In our recent publication, Poster Programs — The Valley of Adventure, some readers are experiencing difficulty with the machine code listing that more experienced users would not come accross.

To make the machine code easier to RUN and test (and harder to crash!), our author, Ray Elder, suggests you make the following changes:

2 REM followed by 41 zeros (instead of 42)
3 REM followed by 240 zeros (instead of 239)
4 REM followed by 98 zeros (instead of 103)
In addition change line 3040 to read 3040 LET F = USR 16985.

To make testing Figure 5 hex dump easier, enter PRINT USR 16985. S$ is an empty string, 32 bytes long ie, 32 spaces. It is used to clear a line. All the machine code and BASIC listings work as published.

Seriously...

A series of programs have been announced by Asher Kuehn which will turn your ZX81 into a very clever business machine.

Among those available are Matrix Planner, a matrix-based modelling and planning 'spreadsheet' program; Wordprocessor, which allows you to process text in capitals or upper and lower case characters on screen or printer; Commodity Trader, a program to help you keep a record of your trades on the commodity market with your broker; Forecast, which provides the user with market forecasts, price fluctuations and practical statistics; and three practical modelling programs based on the Matrix Planner: DCF Planner, Cash-Flow Planner and Profit-Planner.

Alongside these programs, Asher Kuehn have also announced a set of mathematical and statistical programs such as Mathroutines and Fit, which can solve mathematical, technical and scientific problems; RPN Calculator, which simulates the operation of a technical calculator utilising Reverse Polish Notation; and Statistics, which covers t-statistics, chi-squared for equal and unequal expected values and 2 x K contingency tables.

And if these programs seem a little too serious for you, there are three programs which tackle the lighter side of computing. The first, Truth, is a program which enables your ZX81 to enunciate an endless succession of remarks about computing and life in the world of the computer. Using a sophisticated linguistic model to generate English sentences at random, most of its comments are amusing, all are different.

The other two programs available are Adventure and Death, a trip through a medieval castle inhabited by monsters, ghosts and treasure; and Scout 1, which puts you in the cockpit of your own spacecraft, escaping from the alien hordes.

All of these programs are priced between £5 and £10 and are available via mail order. For more information on how to obtain these cassettes, get in touch with Asher Kuehn, a division of Karmead Ltd, 60 St Leonard's Gardens, Heston, Middlesex.

Soft options

A new range of software is now available from Video Software covering the fields of business, education and recreation.

Written for the 16K ZX81, the range includes Video-Sketch, a program exploiting the full graphics capabilities of the ZX81; Video-Map, a game in which you have to navigate your 'plane to its target without being shot down; Video-View, which allows the user to create a miniature view-data system; Video-Graph, a planning and design aid; Video-Index, a sophisticated indexing system allowing up to 1,000 references; Video-Plan, an electronic planning chart or 'spreadsheet'; and Video-Ad, providing an active information display. Video-View, Video-Graph and Video-Ad can also be used on the ZX80 should you possess the necessary amount of memory.

Also available are a number of games cassettes including Football-League, Test-Match, Stock-Market and Party-Tricks.

Every program is recorded twice and with the exception of Video-Index and Party-Tricks, an audio commentary describing the operation of the program is also included. All the more 'serious' programs are accompanied by a comprehensive operating manual.

A similar range of programs will soon be available for the ZX Spectrum.

All of these programs can be ordered by mail order and are priced between £3.95 and £9.95. For more information you could write to Video Software Ltd, Stone Lane, Kinver, Stourbridge, West Midlands DY7 6EQ or 'phone them on 038-483 2462.

Club corner



Following last issue's list of user clubs, we have had a tremendous response from some of the clubs we didn't mention (mainly because we didn't know they existed).

If you run, or are a member of, a user club which caters for the Sinclair user, why not get your group on the map by writing to us at:

Club corner,
ZX Computing,
145 Charing Cross Road,
London WC2H 0EE.

All you have to do is to send us a letter with details of your club (times of meetings, addresses of who to contact, etc) and we'll do the rest. if you publish a newsheet or club magazine, we'd very much like to see that too.

Irish Amateur Computer Club

Dear ZX Computing,
Our recently formed group, the Irish Amateur Computer Club, wish to hear from Sinclair ZX users in the Dublin area and other parts of Ireland.

Those interested should contact either Brendan Haligan at 22 Gortmore Avenue, Finglas Sth., Dublin 11 or myself at the address below. A stamped addressed envelope would be appreciated with all enquiries.
Yours faithfully,

Martin Stapleton,
48 Seacourt,
Clontarf,
Dublin 3.





ZX Exchange

Dear ZX Computing,
ZX Exchange offers opportunities for people to make informal postal contact with ZX users in the UK and abroad. This is especially relevant to those who live in rural areas or who have no local user group.

Full details and a copy of the current ZX Broadsheet are obtainable by sending a stamped addressed envelope and an additional 10p stamp to me at the address below.
Yours faithfully,

Nick Godwin,
4 Hurkur Crescent,
Eyemouth,
Berwickshire TD14 5AP.

Edinburgh ZX Computer Club

Dear ZX Computing,
The Edinburgh ZX Computer Club was formed in October 1981 out of the mutual interest of owners of Sinclair ZX80 and ZX81 computers. Founded by John Palmer and myself, the membership has expanded rapidly and currently stands at over 70. Naturally, the scope of the club now encompasses the ZX Spectrum.

The club provides a chance to meet other ZX owners socially and to exchange ideas and experiences. To this end, various club activities are run. Meetings are held every second and fourth Wednesday of each month in the Claremont Hotel, Claremont Crescent, Edinburgh, from about 7.30 to 10.30pm.

At these meetings, members can bring their computers along, meet other members and help solve each other's problems. *There are* also tutorial groups so that the

more experienced members can pass their knowledge on. Currently, tutorials are being held on beginner's BASIC, advanced BASIC, beginner's machine code, and advanced machine code.

We also publish a bimonthly newsletter which carries news of the club's activities and articles and programs written by some of our members. Another feature of the club are the occasional 'workshops' we hold on Saturdays.

Membership rates are £5 per annum, or £3 for children, students, OAPs and the unemployed.

For more information contact John Palmer, Chairman of the club, at 56 Meadowfield Drive, Edinburgh (Tel: 031-661 3183) or myself at the address below.

Yours faithfully,

Keith Mitchell,
Club Secretary,
19 Meadowplace Road,
Edinburgh EH12 7UJ.
Tel: 031-334 8483.

Swindon Users' Club

Dear ZX Computing,

A users' club has been formed recently in Swindon especially for ZX80, ZX81 and ZX Spectrum users.

We hope to hold monthly meetings and run a software library for both copyright and non-copyright material. For more information on the club contact me at the address below.

Your faithfully,

Andrew Bartlett,
47 Grosvenor Road,
Swindon,
Wilts.
Tel: 0793 30770.

International ZX Spectrum Club

Dear ZX Computing,

Our club will produce a bi-monthly magazine with software, hardware, reviews, contact addresses of other users and news. The objective of the club is to circulate programs around the world.

For further details of the club send an International reply coupon to the following address:

International ZX Spectrum Club,
Gabriel Indalecio Cano
Sardana,
No 4 ático 2a,
San Andrés de la Barca,
Barcelona,
Spain.



The Association of London Computer Clubs

Dear ZX Computing,

The Association of London Computer Club (ALCC) was formed in 1980 as a result of the First London Computer Fair when the North London Hobby Computer Club (NLHCC) invited other clubs to join them in organising this event.

The Association was formed to provide a forum for the Computer Clubs in and around London to enable the clubs to assist each other and to co-operate in areas of common interest. The ALCC aims to promote hobby and recreational computing, co-ordinate the varied activities of the clubs and to organize exhibitions, seminars and meetings.

There are now 16 clubs and the chairman and secretary from each club will together form a clubs council. The main work of the ALCC will now be carried out by 10 specialist sub-committees concentrating on the above-mentioned areas of interest.

The list of computer club members include:

Croydon Micro Computer Club — Meetings are held in the Central Reference Library, Katherine Street on the first and fourth Tuesday of each month. For further information contact Vernon Gifford on 01-653 3207 or David Annal on 01-764 4043.

Harrow Computer Club — Meetings are held at the Harrow College of Higher Education, Room G43 on alternate Wednesdays at 7pm. For further information get in touch with Bazyle Butcher on 01-950 7068.

North London Hobby Computer Club — Meetings are held at the Polytechnic of North London, Holloway Road, every Monday, Tuesday, Wednesday and Thursday during term time. For further information 'phone 01-607 2789 ext. 2161.

East London Amateur Computer Club — Meetings are held at the Harrow Green Library, Cathall Road, on the second and fourth Tuesday of each month at 7pm. For further details contact Fred Linger on 01-554 3288.

North Kent Amateur Computer Club — Meetings are usually held at Charles Darwin School, Biggin Hill, on the first Thursday of the month. For further information contact Barry Biddles on Biggin Hill 71742.

Richmond Computer Club — Meetings are held at the Richmond Community Centre, Sheen Road, on the second Monday of each month at 8pm. For further information get in touch with Robert Forster on 01-892 1873.

South East London Microcomputer Club — Meetings are held at Thames Polytechnic, Woolwich, every other Wednesday at 7pm. Further details are available from Peter Philips on 01-853 5829.

West London Personal Computer Club — Meetings are held at the Fox and Goose, Hanger Lane, on the first Tuesday of each month at 7.45pm. For more information contact either Graham Brian on 01-997 8986 or Neil Cryer on 01-997 9437.

Worcester Park Computer Club — Meetings are held in the Windsor Road Library on the first Monday of each month at 7.30pm. Further details of the club may be obtained from the library on 01-337 1609.

Other Clubs include: Post Office Headquarters (BT & PO), ICPUG(SE), Metropolitan Police, Home Office, ITN, Guildhall and BASUG. Clubs in the following areas are also being organised: Wandsworth, Sutton, the Isle of Dogs and Westminster.

Yours faithfully,

Robin Bradbeer,
Polytechnic of North London,
London N7.



V & H COMPUTER SERVICES PRESENTS

SPECTRUM SPECTACULAR

(50 Programs for the Sinclair Spectrum)

And what programs! New games! Old games! Business programs (like invoicing and sales ledger), word processor, machine code routines, graphics....

All in **SPECTRUM SPECTACULAR** the new book by Roger Valentine.

SPECTACULAR VALUE AT £4.95

By the same author:—

WHAT CAN I DO WITH 1K? (ZX81)......... £4.95
WHAT CAN I DO WITH 16K? (ZX81)....... £4.95

V & H COMPUTER SERVICES
182c KINGSTON ROAD,
STAINES, MIDDX.
TEL: STAINES 58041

Business users please note: Our PAYROLL program is now available for ZX81, SPECTRUM, PET, BBC and KONTRON. Please write for details or send £2.00 for comprehensive manual.

NEW

From Warp Factor Eight



Lift-off into '83 with HI-STAK

ONLY £3·95 includes VAT p&p

- makes your computing easier, faster, more reliable, less exhausting and simply more enjoyable.
- angles your computer neatly to the correct ergonomic position for the most efficient keyboard operation.
- provides extra ventilation for hot ZX81's and Spectrums.
- instantly applied.
- styled for '83, adds that professional look to your set up.
- precision injection moulded in quality ABS, non scratch - non slip base.

ZX81
SPECTRUM
VIC20
TRS80
NEW BRAIN
JUPITER ACE
Etc.

Order form. Please send me ____ HI-STAK set(s) (at £3.95 per set). I enclose cheque/P.O. for ____________

Name ____________

Address ____________

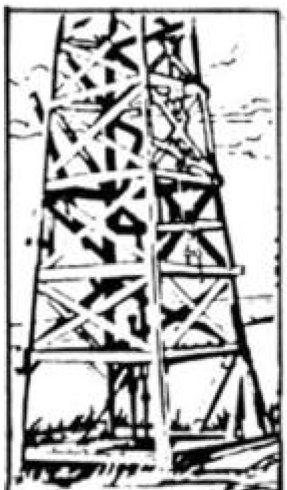
____

____________ Post code ____________

To: Warp Factor Eight, Dept. ZXC.
6 Pelham Road, Braughing, Ware, Herts. SG11 2QU.

C.C.S. MANAGEMENT GAMES

16K ZX81 Spectrum



ALL PROGRAMS AVAILABLE FOR 16K ZX81, SPECTRUM 16K and 48K

NEW

DALLAS

A game of oil exploration and exploitation in Texas. Decisions are required on purchasing concessions, the movement and use of drilling rigs, and the building of platforms and pipelines. Can you take-over the Ewing Empire, or will you be taken-over in the attempt!

AUTOCHEF

As MD of a Catering Company you must negotiate for leases, decide on menu prices, level of wages, advertising and dividends. Each year you must predict the inflation rate. You will be made to resign if you are not successful. There are 3 levels of difficulty.

AIRLINE

Can you compete with British Airways? You must decide on the number of aircraft to operate, staffing and maintenance. Problems encountered are tax demands, strikes, cancelled flights, hijacks and aircraft crashes.

PRINT SHOP

You own a small printing company and are required to decide on (a) the number and type of staff you emloy and when to increase or reduce staff (b) the amount and type of paper you stock (c) the week in which work is scheduled (d) the quotation for each. There are 3 scales of difficulty.

FINANCIAL MODEL — MODELLER X

This is a user friendly business modeller which provides Managers with a tool to plan the strategic decisions of a business. The model shows the profit sensitivity and the interaction of the pertinent marketing and production factors of a business. A user manual is included in the price of the program.



EXCELLENT ROYALTIES PAID FOR SINCLAIR SOFTWARE

PRICES

ZX81 OR SPECTRUM	16K	48K*
Dallas, Airline, Autochef, Print Shop	£5	£6
Modeller X	£8	£9

*Extended 48K Spectrum now available.

MAIL ORDER DISCOUNT

£1.50 for two, £3 for three, £5 for four and £7 for the five.

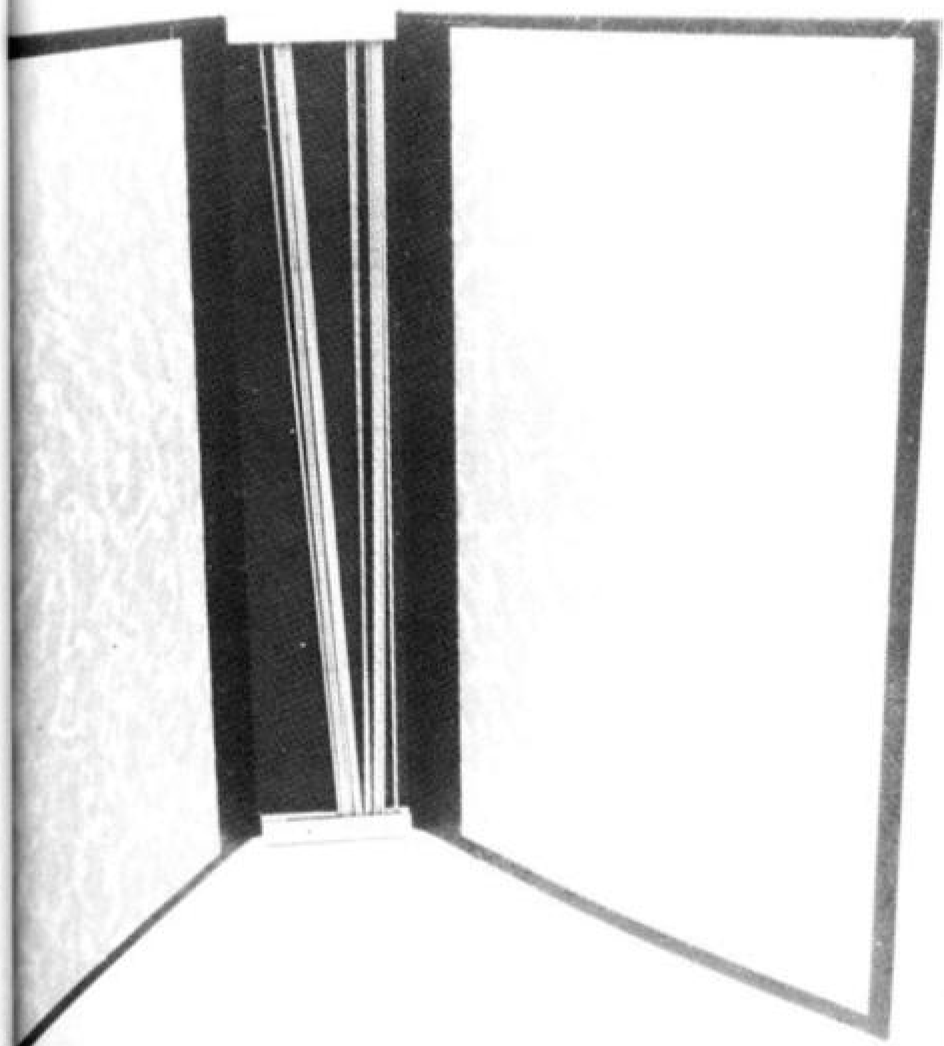
TO ORDER:

Please state computer type and send cheque to:
DEPT ZX, CASES COMPUTER SIMULATIONS
14 LANGTON WAY, LONDON SE3 7TL.

RETAIL STOCKISTS

LONDON	Data-assette, 44 Shroton Street, N.W.1. Buffer Micro Shop, 310 Streatham High Road, S.W.16.
BATH	Micro Style, 29 Belvedere, Lansdown Road.
BOURNEMOUTH	Zedextra, 5 School Lane, Kinson.
DAVENTRY	Micro Style, 67 High Street.
LEICESTER	Microware, Unit 5, St Peters Lane.
MANCHESTER	Micro-Link, 830 Hyde Road. J.M. Computers, 136 Park Lane, Whitefield.
NEWBURY	Micro Style, 47 Cheap Street.

SAVE £4.25 by receiving a FREE magazine binder when you subscribe to ZX COMPUTING



It's true, by subscribing to ZX Computing you will not only receive your personal copy direct to your door for a whole year but also have a superb A.S.P. magazine binder in which to keep your copies AND IT'S FREE!

All you have to do to receive your FREE binder is book a new subscription or renew an existing subscription to ZX Computing before 30th April 1983. What could be simpler? Normally priced at £4.25, these attractive binders will hold approximately 12 issues of ZX Computing.

Not for you any longer the chore of having to track down dogeared copies of your favourite magazine, instead you will merely go to your bookshelf and they will be waiting for you in pristine condition.

Don't miss out on this outstanding offer — subscribe today and receive your FREE binder within 14 days of your order being received.

Cut out and SEND TO :

ZX COMPUTING

513, LONDON ROAD, THORNTON HEATH, SURREY, CR4 6AR ENGLAND.

Please send my free binder and commence my personal subscription with the . . . . issue

SUBSCRIPTION RATES (tick □ as appropriate)

- £12.10 for 6 issues UK □
- £13.60 for 6 issues overseas surface □
- £24.40 for 6 issues Air Mail □

SPECIAL OFFER

I am enclosing my (delete as necessary) Cheque/Postal Order/International Money Order for £. . . . .
(made payable to ASP Ltd)
OR
*Debit my Access/Barclaycard**
*(*delete as necessary)*

Insert card no. □□□□□□□□□□□□□□□□

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss) . . . .
(delete accordingly)

Address . . . .

. . . .

. . . .

Signature . . . .

Date . . . .

Sums and fun for infants

We present an incredible educational software package written for us by Mr P E Bloxham of Loughborough.

This program is written for the ZX81 and occupies 14½K of memory so you'll need to use a 16K RAM pack. Designed for 5-6 year olds, the program comprises a mixture of simple arithmetic tasks and easy-to-play games, the sequence of which is purely random, offering interest and variety to help stimulate the child's learning and retain the child's attention for a reasonable length of time.

When you've typed the program in, you will be greeted with a friendly 'Hello, my name is ZX81, what is your name' to which you will reply with your name. The program then takes you through a series of simple teaching tests followed by a number of games if you manage to get the answers right.

Among the teaching tests are 'Biggest and Smallest' where you have to identify which of two numbers is biggest and which is smallest; 'Simple Sums' in which you are encouraged to attempt basic addition and subtraction; and 'Sort the Numbers' in which you have to sort six numbers into their correct order of magnitude. Where the child is unable to enter the correct answer, after a number of attempts, the correct answer is given. After all entries, except in the games, Newline should be used.

There are also a couple of programs which produce nice effects on the screen. These are 'Fun Time' which draws shapes on the screen such as flowers, spirals and circles; and 'Sign Writer' in which you type your name and the letters are made up into a sign which is moved across the screen.

The games provided in the package are simple, but give a break from the 'number crunching'. Some are offered as reward for correct answers to the sums, while others will be offered to the child quite randomly. There is a bias, however, towards the sums rather than the games, so the child should spend more time involved in the former. The games programs include 'Star Crash' in which you must dodge the falling stars; 'Racing Car' in which you guide a racing car around a track without hitting the sides; and 'Catch a Falling Star' in which you move along the bottom of the screen and try to catch the stars which fall from the top of the screen. The games all have simple instructions and are very user friendly.

There are also some very nice programming touches which should appeal to younger users. For example, should you get the answer to one of the problems wrong you will get a large frowning face on the screen asking you to try the problem again. Should you get the answer right, you are rewarded with a smiling face and a congratulations message.

The program runs at a fairly slow pace so as not to leave the child behind, but can be speeded up if necessary, by adjusting the FOR . . . NEXT loops – used in preference to the PAUSE statement which gives that distracting screen flicker. Likewise, the sums have been kept simple, but could be made more difficult by extending the range of random numbers used.

Initially, children may need a little help from parents but very soon, at the highly receptive age of 5 or 6 years, they will be more than able to use this program pretty much unaided. This program presents a complete package for the younger ZX user, being both fun to use as well as educational.



```
   1 REM
   2 REM SUMS + FUN FOR INFANTS
   3 REM
   4 GOTO 9800
  10 REM BIGGEST+SMALLEST TEST
  11 REM
  12 PRINT AT 8,5;"BIGGEST AND SMALLEST"
  15 PRINT AT 9,5;"*******         ********"
  16 PRINT AT 10,5;"****************"
  20 PRINT AT 14,5;"QUIZ FOR ";A$
  25 PRINT AT 15,5;"*******************"
  30 FOR I=1 TO 200
  35 NEXT I
  40 CLS
  45 RAND
  50 LET X=INT (1+30*RND)
  55 LET Y=INT (1+30*RND)
  60 LET Z=INT (1+30*RND)
  61 IF X=Y THEN GOTO 50
  62 IF X=Z THEN GOTO 50
  63 IF Y=Z THEN GOTO 50
  65 LET P=INT (1+10*RND)
  70 IF P<6 THEN GOTO 150
  80 REM BIGGEST
  81 REM
  82 LET B=0
  85 PRINT TAB 3;"                ******  ***"
  90 PRINT TAB 3;"WHICH IS*BIGGEST*";A$;"?"
  95 PRINT TAB 3;"                ******  ***"
  96 PRINT
  97 PRINT ""
  98 PRINT "    ";X;"        ";Y;"    ";Z
  99 PRINT ""
 100 IF B<X THEN LET B=X
 110 IF B<Y THEN LET B=Y
 120 IF B<Z THEN LET B=Z
 125 INPUT B1
 130 REM B=ACTUAL BIGGEST
 131 REM
 132 REM B1=CHOSEN BIGGEST
 133 REM
 140 IF B1=B THEN GOTO 9200
 145 GOSUB 9100
```

```
146 GOTO 80
150 REM SMALLEST
151 REM ████████
155 LET B=31
160 PRINT TAB 3;"           ******
****"
165 PRINT TAB 3;"WHICH IS*SMALL
EST*";A$;"?"
170 PRINT TAB 3;"           ******
****"
175 PRINT
180 PRINT "██████████████████████
██████"
185 PRINT "    ";X;"          ";
Y;"      ";Z
190 PRINT "██████████████████████
██████"
195 IF B>X THEN LET B=X
200 IF B>Y THEN LET B=Y
205 IF B>Z THEN LET B=Z
210 REM B=ACTUAL SMALLEST
211 REM ████████████████████
212 REM B1=CHOSEN SMALLEST
213 REM ████████████████████
215 INPUT B1
220 IF B1=B THEN GOTO 9200
225 GOSUB 9100
230 GOTO 150
1000 REM SIMPLE SUMS
1001 REM ███████████
1008 PRINT TAB 9;"***********"
1009 PRINT TAB 9;"***********"
1010 PRINT TAB 9;"SIMPLE SUMS"
1015 PRINT TAB 9;"***********"
1016 PRINT TAB 9;"***********"
1020 PRINT
1025 PRINT
1030 PRINT
1035 PRINT TAB 10;"NOW ";A$
1040 PRINT
1045 PRINT
1050 PRINT TAB 6;"LET US DO SOME
SUMS"
1060 FOR I=1 TO 200
1065 NEXT I
1070 CLS
1075 RAND
1080 LET X=INT (1+20*RND)
1090 LET Y=INT (1+20*RND)
1100 LET A=INT (1+10*RND)
1105 IF A<6 THEN GOTO 1360
1110 REM ADDITION
1111 REM ████████
1115 LET T=0
1120 FOR I=1 TO 7
1130 PRINT
1140 NEXT I
1150 PRINT "██████████████████████
██████"
1160 PRINT
1170 PRINT TAB 7;X;"   +   ";Y;"
=  ???"
1180 PRINT
1190 PRINT "██████████████████████
██████"
1200 INPUT Z
1210 LET T=T+1
1220 IF T=4 THEN GOTO 1250
1223 CLS
1225 IF Z=X+Y THEN GOTO 9200
1230 GOSUB 9100
1240 GOTO 1150
1250 PRINT
1260 PRINT "WELL ";A$
1270 PRINT "I WILL HELP YOU NOW"
1280 PRINT "THE ANSWER IS ";X+Y
1290 PRINT "               ****"
1300 FOR I=1 TO 100
1310 NEXT I
1350 GOTO 9035
1360 REM SUBTRACTION
1365 REM ███████████
1370 IF X<Y THEN GOTO 1070
1380 LET T=0
1390 FOR I=1 TO 7
1400 PRINT
1410 NEXT I
1420 PRINT "██████████████████████
██████"
1430 PRINT
1440 PRINT TAB 7;X;"   -   ";Y;"
=  ???"
1450 PRINT
1460 PRINT "██████████████████████
██████"
1470 INPUT Z
1480 LET T=T+1
1490 IF T=4 THEN GOTO 1530
1495 CLS
1500 IF Z=X-Y THEN GOTO 9200
1510 GOSUB 9100
1520 GOTO 1420
1530 PRINT
1540 PRINT "WELL ";A$
1550 PRINT "I WILL HELP YOU NOW"
1560 PRINT "THE ANSWER IS ";X-Y
1570 PRINT "               ****"
1580 FOR I=1 TO 100
1585 NEXT I
1590 GOTO 9035
2000 REM GAME-STARCRASH
2001 REM ████████████
2010 PRINT "LET US PLAY A GAME "
;A$
2011 PRINT
2012 PRINT
2013 PRINT
2014 PRINT
2015 PRINT TAB 11;"STARCRASH"
2016 PRINT TAB 11;"*********"
2017 PRINT TAB 10;"***********"
2018 PRINT
2019 PRINT
2020 PRINT "PRESS M TO STEER YOU
R SPACESHIP"
2021 PRINT "        █"
2022 PRINT
2023 PRINT "IF YOU HIT A STAR TH
E GAME ENDS"
2024 PRINT "AND I WILL TELL YOU
YOUR SCORE"
2025 PRINT
2026 PRINT
2027 PRINT
2030 PRINT "GET READY-THE GAME S
TARTS SOON"
2040 FOR I=1 TO 120
2045 NEXT I
2050 CLS
2060 LET W=0
2070 LET X=10
2080 LET Y=15
2090 LET Z=20
2100 PRINT AT Z,RND*30;"*"
2110 PRINT AT X,Y;"   "
2120 LET W=W+1
2130 SCROLL
2140 IF Y>2 THEN LET Y=Y-1
2150 IF INKEY$="M" AND Y<28 THEN
 LET Y=Y+2
2160 PRINT AT X,Y;"▀▄▀"
2170 PRINT AT 11,Y+1;
2180 IF PEEK (PEEK 16398+256*PEE
K 16399)<>23 THEN GOTO 2100
2190 PRINT W
2200 FOR I=1 TO 35
2205 NEXT I
2210 CLS
2220 PRINT
2221 PRINT
2222 PRINT
2223 PRINT
2225 IF W>50 THEN GOTO 2280
2230 PRINT "NOT BAD ";A$
2231 PRINT "YOUR SCORE WAS..";W
2235 PRINT
2240 PRINT "WAS THAT FUN?"
2245 PRINT
2250 PRINT "LET US DO SOMETHING
ELSE NOW"
2260 FOR I=1 TO 45
2265 NEXT I
2270 GOTO 9035
```

```
2280 IF W>300 THEN GOTO 2296
2290 PRINT "WELL DONE ";A$
2295 GOTO 2231
2296 PRINT "VERY GOOD INDEED ";A
$
2298 GOTO 2231
3000 REM FUN--TIME
3001 REM
3010 PRINT TAB 6;"FUN--TIME ";A$
3015 PRINT TAB 4;"**************
*****"
3016 PRINT TAB 4;"**************
*****"
3020 PRINT
3021 PRINT
3022 PRINT
3030 PRINT "WATCH ME NOW ";A$
3035 PRINT
3040 PRINT "AND I WILL DRAW A PR
ETTY"
3045 PRINT
3050 PRINT "SHAPE FOR YOU"
3055 FOR I=1 TO 80
3058 NEXT I
3060 CLS
3065 RAND
3070 LET A=INT (1+10*RND)
3080 IF A<4 THEN GOTO 3200
3090 IF A>6 THEN GOTO 3400
3100 REM PRETTY SHAPE
3105 REM
3110 FOR X=1 TO 100
3120 LET Y=PI*X/50
3130 PRINT AT 9*COS (Y)+10,14*SI
N (Y)+15;"■"
3140 NEXT X
3150 FOR I=1 TO 15
3155 NEXT I
3160 GOTO 3300
3200 REM FLOWERS+SPIRALS
3201 REM
3205 LET A=INT (1+10*RND)
3208 IF A>5 THEN GOTO 3260
3210 REM PRETTY FLOWER
3211 REM
3215 FOR S=0 TO 200 STEP 3
3220 LET T=S*PI/180
3225 LET F=23*SIN (5*T)
3230 LET P=30+F*COS T
3235 LET Q=20+F*SIN T
3240 PLOT P,Q
3245 NEXT S
3250 FOR I=1 TO 15
3253 NEXT I
3255 GOTO 3300
3260 REM PRETTY SPIRAL
3261 REM
3265 FOR S=0 TO 1300 STEP 8
3270 LET T=S*PI/180
3275 LET F=1.1*T
3280 LET P=30+F*COS T
3285 LET Q=20+F*SIN T
3290 PLOT P,Q
3295 NEXT S
3296 FOR I=1 TO 15
3298 NEXT I
3300 CLS
3310 PRINT AT 8,5;"NOW ";A$
3320 PRINT AT 10,5;"ONTO SOMETHI
NG ELSE"
3325 FOR I=1 TO 35
3328 NEXT I
3330 GOTO 9035
3400 REM PRETTY CIRCLES
3405 REM
3410 FOR J=10 TO 2 STEP -1
3415 LET Q$=CHR$ (INT (RND*11+12
8*(RND<0.5)))
3420 FOR B=0 TO 360 STEP 10
3430 LET G=B*PI/180
3440 PRINT AT 10+J*COS G,15+J*SI
N G;Q$
3450 NEXT B
3460 NEXT J
3470 FOR I=1 TO 15
3475 NEXT I
3480 GOTO 3300
4000 REM SORT THE NUMBERS
4001 REM
4005 PRINT TAB 6;"**************
**"
4010 PRINT TAB 6;"SORT THE NUMBE
RS"
4015 PRINT TAB 6;"**************
**"
4020 FOR X=1 TO 5
4025 PRINT
4030 NEXT X
4035 PRINT "WHEN I PUT 6 NUMBERS
 ON"
4036 PRINT
4040 PRINT "THE SCREEN,YOU SORT
THEM"
4041 PRINT
4045 PRINT "INTO ORDER-SMALLEST
TO BIGGEST"
4050 PRINT
4055 PRINT
4060 PRINT "IF YOU GET YOUR ANSW
ER RIGHT"
4070 PRINT
4075 PRINT "I WILL LET YOU PLAY
A GAME"
4080 PRINT
4085 PRINT "I AM CHOOSING THE NU
MBERS NOW..."
4090 FOR I=1 TO 80
4093 NEXT I
4095 CLS
4100 REM SELECTION OF 6 NUMBERS
4101 REM
4105 RAND
4110 DIM A(6)
4115 FOR I=1 TO 6
4120 LET A(I)=INT (1+20*RND)
4121 NEXT I
4122 IF A(2)=A(1) THEN GOTO 4100
4123 IF A(3)=A(2) OR A(3)=A(1) T
HEN GOTO 4100
4124 IF A(4)=A(3) OR A(4)=A(2) O
R A(4)=A(1) THEN GOTO 4100
4125 IF A(5)=A(4) OR A(5)=A(3) O
R A(5)=A(2) OR A(5)=A(1) THEN GO
TO 4100
4126 IF A(6)=A(5) OR A(6)=A(4) O
R A(6)=A(3) OR A(6)=A(2) OR A(6)
=A(1) THEN GOTO 4100
4130 PRINT "NOW ";A$
4131 PRINT
4135 PRINT "MY NUMBERS ARE......
..........."
4140 PRINT
4145 FOR I=1 TO 6
```



```
4150 PRINT A(I);" ";
4155 NEXT I
4158 LET C=0
4160 REM A$ SORT+ENTER NUMBERS
4161 REM [redacted]
4165 PRINT
4166 PRINT
4170 PRINT "YOU ENTER THE NUMBER
S NOW"
4171 PRINT
4175 PRINT "SMALLEST FIRST,TO BI
GGEST LAST"
4177 PRINT
4180 DIM B(6)
4185 FOR I=1 TO 6
4190 INPUT B(I)
4195 PRINT B(I);" ";
4196 IF B(I)=A(1) OR B(I)=A(2) O
R B(I)=A(3) OR B(I)=A(4) OR B(I)
=A(5) OR B(I)=A(6) THEN GOTO 419
8
4197 GOTO 4500
4198 NEXT I
4199 FOR I=1 TO 12
4200 NEXT I
4201 REM ZX SORT THE NUMBERS
4202 REM [redacted]
4203 LET K=0
4205 FOR I=1 TO 5
4208 IF A(I)>A(I+1) THEN GOTO 42
15
4210 GOTO 4230
4215 LET T=A(I)
4220 LET A(I)=A(I+1)
4225 LET A(I+1)=T
4226 LET K=K+1
4230 NEXT I
4231 IF K=0 THEN GOTO 4250
4232 GOTO 4200
4250 REM COMPARE ZX+A$ SORTS
4251 REM [redacted]
4255 FOR I=1 TO 6
4260 IF A(I)<>B(I) THEN GOTO 430
0
4265 NEXT I
4270 REM CORRECT ANSWER GIVEN
4271 REM [redacted]
4272 PRINT
4273 PRINT
4274 PRINT
4275 PRINT TAB 5;"***WELL DONE "
;A$;"***"
4278 PRINT
4280 PRINT TAB 5;"NOW YOU CAN PL
AY A GAME"
4281 PRINT "********************
***********"
4285 FOR I=1 TO 35
4288 NEXT I
4290 CLS
4295 GOTO 8000
4300 REM WRONG ANSWER GIVEN
4301 REM [redacted]
4310 LET C=C+1
4315 IF C=3 THEN GOTO 4400
4320 CLS
4325 PRINT "NO ";A$
4327 PRINT
4330 PRINT "NOT QUITE RIGHT"
4335 PRINT
4340 PRINT "YOUR NUMBERS WERE...
..........."
4345 PRINT
4350 FOR I=1 TO 6
4355 PRINT B(I);" ";
4360 NEXT I
4365 PRINT
4366 PRINT
4370 PRINT "NOW TRY AGAIN"
4375 GOTO 4160
4400 REM HELP NOW NEEDED
4401 REM [redacted]
4405 CLS
4410 PRINT "WELL ";A$
4415 PRINT "I WILL HELP YOU NOW"
4420 PRINT
4421 PRINT
4425 PRINT "YOUR NUMBERS WERE...
..........."
4430 PRINT
4435 FOR I=1 TO 6
4440 PRINT B(I);" ";
4445 NEXT I
4450 PRINT
4451 PRINT
4455 PRINT "FROM SMALLEST TO BIG
GEST"
4460 PRINT "THE NUMBERS SHOULD B
E..........."
4465 PRINT
4470 FOR I=1 TO 6
4475 PRINT A(I);" ";
4480 NEXT I
4485 PRINT
4486 PRINT
4489 PRINT "THERE YOU ARE ";A$
4490 PRINT "CHECK WHERE YOU WENT
WRONG"
4491 PRINT "********************
***********"
4492 FOR I=1 TO 75
4493 NEXT I
4494 PRINT
4495 PRINT "NOW TRY SOME SIMPLE
SUMS"
4496 FOR I=1 TO 15
4497 NEXT I
4498 CLS
4499 GOTO 1000
4500 REM INVALID NUMBER INPUT
4501 REM [redacted]
4510 PRINT
4520 PRINT "WRONG NUMBER-NOT IN
MY LIST"
4521 PRINT "********************
*******"
4525 PRINT
4530 PRINT "LET US START AGAIN"
4540 FOR I=1 TO 35
4545 NEXT I
4550 CLS
4560 GOTO 4100
8000 REM REWARD GAMES
8001 REM [redacted]
8005 RAND
8010 LET G=INT (1+10*RND)
8015 IF G<4 THEN GOTO 8400
8020 IF G>7 THEN GOTO 8600
8100 REM RACING CAR
8110 REM [redacted]
8115 PRINT TAB 10;"▚▚▚▚▚▚▚▚▚▚"
8120 PRINT TAB 10;"RACING CAR"
8130 PRINT TAB 10;"▚▚▚▚▚▚▚▚▚▚"
8131 PRINT
8132 PRINT
8133 PRINT
8134 PRINT
8135 PRINT "RACE YOUR CAR ROUND
THE TRACK"
8137 PRINT
8140 PRINT "BUT DO NOT HIT THE S
IDES"
8145 PRINT
8150 PRINT "OR THE GAME WILL END
"
8155 PRINT
8156 PRINT
8157 PRINT
8158 PRINT
8160 PRINT "PRESS Z TO GO LEFT"
8165 PRINT "■.."
8170 PRINT "PRESS M TO GO RIGHT"
8175 PRINT "■.."
8176 PRINT
8180 PRINT "THE GAME STARTS SOON
"
8185 FOR I=1 TO 50
8188 NEXT I
8190 CLS
8200 LET W=10
8210 LET X=10
8220 LET Y=20
8230 LET Z=9
```

```
8240 PRINT AT Y,Z;"▓██      ██▓"
8250 PRINT AT W,X;":"
8260 SCROLL
8270 IF INKEY$="Z" THEN LET X=X-1
8280 IF INKEY$="M" THEN LET X=X+1
8290 PRINT AT W,X;"Y"
8300 IF Z<17 THEN LET Z=Z+2*RND
8310 IF Z>7 THEN LET Z=Z-2*RND
8320 PRINT AT 11,X;
8330 IF PEEK (PEEK 16398+PEEK 16399*256)=128 THEN GOTO 8350
8340 GOTO 8240
8350 FOR I=1 TO 15
8355 NEXT I
8360 GOTO 9035
8400 REM SIGN-WRITER
8401 REM ███████████
8403 PRINT TAB 10;"███████████"
8405 PRINT TAB 10;"SIGN-WRITER"
8410 PRINT TAB 10;"███████████"
8415 PRINT
8416 PRINT
8417 PRINT
8420 PRINT "YOU ENTER YOUR FULL NAME"
8425 PRINT
8430 PRINT "AND I WILL CHANGE IT INTO A"
8435 PRINT
8440 PRINT "MOVING SIGN ACROSS THE SCREEN"
8445 INPUT B$
8450 CLS
8455 PRINT AT 1,0;"████████████████████████████████"
8456 PRINT AT 3,0;"▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚"
8457 PRINT AT 5,0;"********************************"
8458 PRINT AT 16,0;"********************************"
8459 PRINT AT 18,0;"▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚"
8460 PRINT AT 20,0;"████████████████████████████████"
8465 FOR I=1 TO 2
8470 LET P=LEN B$
8475 LET C=1
8480 IF C<31 THEN PRINT AT 10,(31-C);B$(1 TO C)
8485 IF C>=31 AND C<=LEN B$ THEN PRINT AT 10,0;B$(C-30 TO C)
8490 IF C>=LEN B$ THEN LET B$=B$+" "
8495 IF LEN B$=P+32 THEN GOTO 8510
8500 LET C=C+1
8505 GOTO 8480
8510 NEXT I
8515 FOR S=1 TO 10
8520 PRINT AT 10,12;"GOODBYE"
8525 PRINT AT 10,12;"GOODBYE"
8530 NEXT S
8535 FOR I=1 TO 30
8538 NEXT I
8540 GOTO 9035
8600 REM CATCH A FALLING STAR
8601 REM ████████████████████
8605 PRINT TAB 5;"********************"
8610 PRINT TAB 5;"CATCH A FALLING STAR"
8615 PRINT TAB 5;"********************"
8620 FOR I=1 TO 5
8625 PRINT
8630 NEXT I
8635 PRINT "  AS THE STAR FALLS FROM THE"
8640 PRINT
8645 PRINT "  SKY,YOU TRY TO CATCH IT"
8650 PRINT
8655 PRINT "  PRESS Z TO GO LEFT"
8660 PRINT "            ▀"
8665 PRINT "  PRESS M TO GO RIGHT"
8670 PRINT "            ▀"
8675 PRINT
8680 PRINT "  THE GAME STARTS SOON"
8685 FOR I=1 TO 35
8688 NEXT I
8690 CLS
8695 LET X=0
8700 LET Y=16
8705 FOR R=1 TO 5
8710 LET S=INT (RND*25)
8715 FOR T=0 TO 20
8720 IF INKEY$="Z" THEN LET Y=Y-1
8725 IF INKEY$="M" THEN LET Y=Y+1
8730 IF Y<0 THEN LET Y=0
8735 IF Y>31 THEN LET Y=31
8740 CLS
8745 PRINT AT T,S;"*";AT 20,Y;"▙▟"
8750 IF T=20 AND S=Y THEN LET X=X+1
8751 IF T=20 AND S=Y+1 THEN LET X=X+1
8755 IF T=20 AND S=Y THEN GOSUB 8850
8760 IF T=20 AND S=Y+1 THEN GOSUB 8850
8765 NEXT T
8770 NEXT R
8775 FOR I=1 TO 10
8778 NEXT I
8780 CLS
8785 PRINT AT 8,0;"WELL, ";A$
8790 PRINT AT 10,0;"YOU CAUGHT ";X;" OUT OF 5 STARS"
8795 PRINT AT 12,0;"NOW LET US DO SOMETHING ELSE"
8800 FOR I=1 TO 35
8803 NEXT I
8805 GOTO 9035
8850 FOR I=1 TO 10
8855 PRINT AT 20,Y;"**"
8860 PRINT AT 20,Y;"**"
8865 NEXT I
8870 RETURN
9000 REM GENERAL INTRODUCTION
9001 REM ████████████████████
9005 PRINT AT 8,3;"HELLO,MY NAME IS ZX81"
9006 PRINT AT 9,3;"*********************"
9015 PRINT AT 12,3;"WHAT IS YOUR NAME?"
9016 PRINT AT 13,3;"******************"
9020 INPUT A$
9021 PRINT
9022 PRINT
9023 PRINT
9025 PRINT "********************************"
9026 PRINT "NOW ";A$
9027 PRINT "LET US HAVE SOME FUN"
9028 PRINT "HERE WE GO......................"
9029 PRINT "********************************"
9030 FOR I=1 TO 35
9033 NEXT I
9035 CLS
9040 REM SELECTION OF ACTIVITIES
9041 REM ███████████████████████
9045 FOR I=1 TO 10
9048 NEXT I
9049 RAND
9050 LET A=INT (1+10*RND)
9055 IF A<7 THEN GOTO 9090
9060 LET B=INT (1+10*RND)
9065 IF B>7 THEN GOTO 2000
9070 LET C=INT (1+10*RND)
```

```
9075 IF C<4 THEN GOTO 3000
9080 GOTO 4000
9090 LET D=INT (1+10*RND)
9095 IF D<6 THEN GOTO 10
9096 GOTO 1000
9100 REM WRONG ANSWER GIVEN
9101 REM ████████████████████
9102 PRINT
9105 PRINT "NO ";A$;",TRY AGAIN"
9110 PRINT
9115 PRINT
9120 PRINT "                  █████████
"
9125 PRINT "                  █       █
"
9130 PRINT "                  █--   --█
"
9135 PRINT "                 ██ O   O █
█"
9140 PRINT "                 ██ : ▒ : █
█"
9145 PRINT "                 ██       █
█"
9150 PRINT "                  █  ▒▒▒  █
"
9155 PRINT "                  █ ▒ ▒   █
"
9160 PRINT "                  █       █
"
9165 PRINT "                  █████████
"
9170 FOR I=1 TO 25
9173 NEXT I
9175 CLS
9180 RETURN
9200 REM CORRECT ANSWER GIVEN
9201 REM ████████████████████
```




```
9202 PRINT
9205 PRINT "WELL DONE ";A$;"-HAV
E A STAR."
9210 PRINT
9215 PRINT
9220 PRINT "                  █████████
"
9225 PRINT "                  █       █
"
9230 PRINT "                  █--   --█
"
9235 PRINT "                 ██ O   O █
█"
9240 PRINT "                 ██   ▒   █
█"
9245 PRINT "                 ██       █
█"
9250 PRINT "                  █ ▒   ▒ █
"
9255 PRINT "                  █ ▒   ▒ █
"
9260 PRINT "                  █ ▒▒▒▒▒ █
"
9265 PRINT "                  █       █
"
9270 PRINT "                  █████████
"
9275 FOR I=1 TO 25
9278 NEXT I
9280 FOR I=1 TO 3
9285 CLS
9290 PRINT
9295 PRINT
9300 PRINT
9305 PRINT
9310 PRINT
9315 PRINT "                  *    *    *
"
9320 PRINT "                   *   *   *
"
9325 PRINT "                    *  *  *
"
9330 PRINT "                      ***"
9335 PRINT "                  *********
"
9340 PRINT "                      ***"
9345 PRINT "                    *  *  *"
9350 PRINT "                   *   *   *"
9355 PRINT "                  *    *    *
"
9360 NEXT I
9365 GOTO 9035
9800 REM OPENING TITLE PAGE
9801 REM ████████████
9805 FOR I=1 TO 5
9810 PRINT
9815 NEXT I
9820 PRINT "████████████████████
███████████"
9825 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒"
9830 PRINT "████████████████████
███████████"
9835 PRINT "********************
************"
9840 PRINT "SUMS AND FUN FOR INF
ANTS"
9845 PRINT "▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
▚▚▚"
9850 PRINT
9855 PRINT
9860 PRINT "BY P.E.BLOXHAM
9865 PRINT "▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
"
9870 PRINT "********************
************"
9875 PRINT "████████████████████
███████████"
9880 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒"
9885 PRINT "████████████████████
███████████"
9886 FOR I=1 TO 35
9887 NEXT I
9888 CLS
9889 GOTO 9000
9900 REM AUTO-RUN
9901 REM ██████
9910 SAVE "S+█"
9920 GOTO 1
9990 REM MEMORY USED
9991 REM ███████████
9992 PRINT PEEK 16396+256*PEEK 1
6397-16509
```



Defending your Spectrum

David Cross of Wolverhampton has written us a great program in which you have to defend the Earth from the invading aliens.

See page 114 for further details

Not an easy game this! You have three ships, and your mission is to rid the planet's surface of invading aliens.

Each time you fire a missile at an invader and destroy it you will be awarded 60 points but watch out, your missiles may not always destroy the aliens. If an invader hits you then you will lose one of your ships. The only way to get more ships is to increase your score; you will get an extra ship when your score increases to 5,000, 10,000, 15,000 and 20,000.

The control keys for the game are as follows:

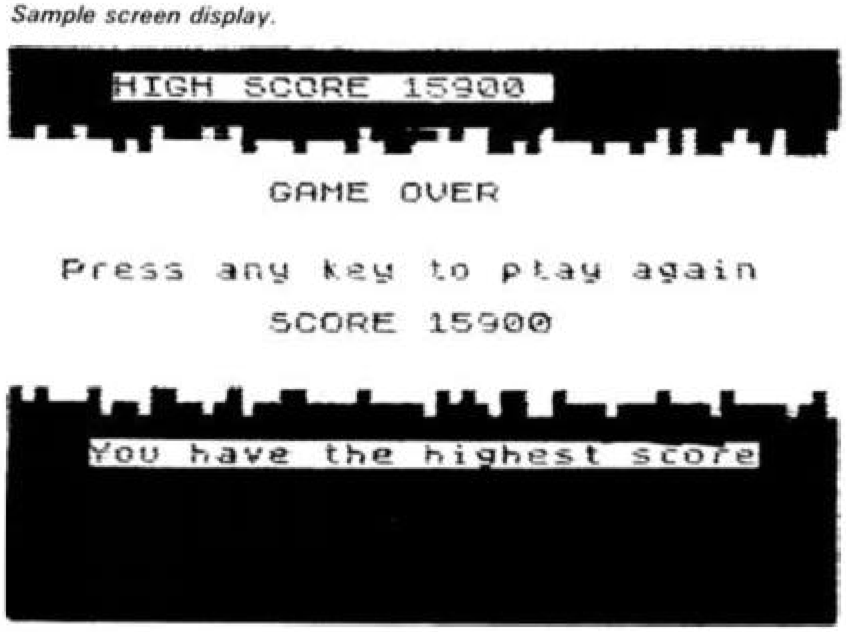
The '1' key moves your ship down
The 'Q' key moves your ship up
The 'P' key fires a missile
The 'T' key stops the game

Once the program has been typed in and RUN, you will have to wait four seconds for the game to start – not long really, when you consider the fate of Earth is in your hands!

```
   3 BORDER 0: PAPER 0: INK 7: C
LS
   4 GO SUB 1000
   7 CLEAR
  10 LET t=0: LET hs=0: LET d=4:
 LET s=0: LET a=10: LET l=3
  20 LET a$="[illegible]"
  30 LET b$="[illegible]"
  40 FOR f=14 TO 21
  50 PRINT INK 4;AT f,0;"[illegible]"
  60 NEXT f
  65 FOR h=0 TO 2
  70 PRINT INK 4;AT h,0;"[illegible]"
  80 NEXT h
  90 PRINT PAPER 0; INK 7;AT 19,
6;" SCORE "
 180 PRINT AT d,0;"           ": LET
 d=RND*6+5: LET i=28
 190 LET f=7: LET t=0
 200 IF t=1 AND INKEY$="q" OR t=
1 AND INKEY$="1" THEN GO TO 190
 201 LET a=a+(INKEY$="q")-(INKEY
$="1")
 202 IF INKEY$="t" THEN GO TO 75
0
 203 IF INKEY$="p" THEN LET t=1
 205 IF a<5 THEN LET a=5
 210 IF a>11 THEN LET a=11
 215 PRINT AT a-1,3;"          "
 216 PRINT AT a+1,3;"          "
 218 PRINT AT a,3;"    "
 220 IF t=0 THEN GO TO 230
 221 IF SCREEN$ (a,f)<>" " THEN
GO TO 500
 222 IF SCREEN$ (a,f+1)<>" " THE
N GO TO 500
 223 PRINT AT a,f;CHR$ 146
 225 BEEP .02,0
 227 PRINT AT a,f;"  "
 229 LET f=f+1
```

```
 230 IF SCREEN$ (a,7)<>" " THEN
GO TO 600
 234 PRINT INK 3;AT a,5;CHR$ 144
;CHR$ 145
 235 PRINT INK 6;AT a,3;CHR$ 149
;CHR$ 146
 240 PRINT PAPER 7;AT 10,10;SCRE
EN$ (a,6)
 250 PRINT PAPER 0; INK 7;AT 19,
13;s;"  "
 260 PRINT INK 4;AT 13,0;a$
 265 PRINT INK 4;AT 3,0;b$
 270 LET a$=a$(1 TO 32)+a$(1)
 275 LET b$=b$(1 TO 32)+b$(1)
 280 LET a$=a$(2 TO )
 285 LET b$=b$(2 TO )
 290 PRINT INK 7:AT d,i;CHR$ 147
;"  "
 295 IF s>5000 THEN LET l=l+1
 296 IF s>10000 THEN LET l=l+2
 297 IF s>15000 THEN LET l=l+1
 298 IF s>20000 THEN LET l=l+1
 300 IF i>=13 THEN LET i=i-(RND*
1+1)
 305 IF i<13 THEN LET i=i-1
 310 IF i<=3 THEN GO TO 180
 315 IF f=31 THEN GO TO 180
 320 GO TO 200
 500 FOR h=1 TO 3
 505 PRINT INK 5;AT a,3;CHR$ 149
;CHR$ 148
 510 INK 7: PRINT AT a-1,f;CHR$
152;CHR$ 153: PRINT AT a,f;CHR$
154;CHR$ 155: PAUSE 10
 520 INK 6: PRINT AT a-1,f;CHR$
156;CHR$ 157: PRINT AT a,f;CHR$
158;CHR$ 159: PAUSE 10
 521 INK 7
 525 BEEP .1,-30: BEEP .1,-25
 530 NEXT h
 537 LET s=s+60
 540 PRINT AT a-1,f;"  ": PRINT
AT a,f;"  "
 550 GO TO 180
 600 FOR g=1 TO 3: INK 5
 602 PRINT AT a-1,4;" ";CHR$ 152
;"■";CHR$ 153
 605 PRINT AT a,4;" ";CHR$ 154;"
■";CHR$ 155: PAUSE 15
 607 INK 6
 608 BEEP .05,-30: BEEP .07,-25:
 BEEP .1,-20
 610 PRINT AT a-1,4;" ";CHR$ 156
;"■";CHR$ 157
 615 PRINT AT a,4;" ";CHR$ 158;"
■";CHR$ 159: PAUSE 10
 620 NEXT g
 627 PRINT AT a-1,4;"     "
 630 PRINT AT a,4;"     "
 632 PRINT AT a+1,4;"     ": PAU
SE 15
 635 BEEP .1,-10: BEEP .07,-4: B
EEP .2,0
 640 LET l=l-1
 650 IF l=0 THEN GO TO 750
 700 GO TO 180
 752 PRINT AT 5,10;"GAME OVER"
 755 FOR f=1 TO 3
 760 BEEP .1,-3: BEEP .1,-15: BE
EP .1,-4: BEEP .1,-5: BEEP .2,5:
 BEEP .1,-20
 770 NEXT f
 775 BEEP .1,-10: BEEP .07,-4: B
EEP .2,0: BEEP .2,2: BEEP .4,-20
 777 IF s>hs THEN LET hs=s
 780 PRINT PAPER 0; INK 7;AT 1,4
;"HIGH SCORE ";hs;" ": PRINT AT
5,10;"GAME OVER": PRINT ;AT 8,2;
"Press any key to play again": P
RINT AT 10,10;"SCORE ";s
 785 PRINT PAPER 4;AT 19,6;"
  "
 786 IF s>=hs THEN PRINT PAPER 0
; INK 7;AT 15,3;"You have the hi
ghest score"
 790 PAUSE 40
 800 PRINT AT 5,10;"         "
 810 PAUSE 30
 820 IF INKEY$<>"" THEN GO TO 7
 830 GO TO 780
1000 FOR n=144 TO 159
1010 FOR f=0 TO 7
1020 READ x: POKE USR CHR$ (n)+f
,x
1030 NEXT f
1040 NEXT n
1050 DATA 0,0,BIN 01111100,BIN 1
1111110,BIN 11111111,BIN 1111111
1,BIN 11111111,BIN 01111100
1060 DATA 0,0,0,0,0,BIN 11111111
,BIN 11111100,0
1070 DATA 0,0,0,0,0,BIN 00111110
,BIN 00111110,0
1080 DATA BIN 00111000,BIN 01010
100,BIN 01111100,BIN 00111000,BI
N 01010100,BIN 10010010,BIN 1001
0010,BIN 10010010
1100 DATA 0,0,0,BIN 00011111,BIN
 11111111,BIN 11111111,BIN 00011
111,0
1110 DATA 0,0,0,0,BIN 00000001,B
IN 00011111,0,0
1120 DATA 0,BIN 11111100,255,255
,255,255,BIN 11111110,BIN 111100
00
1130 DATA 0,0,BIN 10000000,BIN 1
1111000,BIN 11111111,BIN 1110000
0,0,0
1140 DATA 0,BIN 00001111,BIN 000
11111,BIN 00011111,BIN 00111111,
BIN 01111111,BIN 01111111,BIN 01
111111
1150 DATA 0,BIN 10000000,BIN 111
00000,BIN 11111000,BIN 11111100,
BIN 11111100,BIN 11111110,BIN 11
111110
1160 DATA BIN 01111111,BIN 01111
111,BIN 01111111,BIN 00111111,BI
N 00011111,BIN 00001111,BIN 0000
00111,0,255
1170 DATA BIN 11111110,BIN 11111
110,BIN 11111110,BIN 11111100,BI
N 11111000,BIN 11110000,BIN 1110
0000
1180 DATA BIN 00111111,BIN 00111
111,BIN 01111111,255,255,255,255
,255
1190 DATA BIN 11110000,BIN 11111
000,BIN 11111110,255,255,255,255
,255
1200 DATA 255,255,255,255,255,25
5,BIN 00111111,BIN 00011111,255
1210 DATA 255,255,255,255,255,25
5,BIN 11111110,BIN 11111100
1230 RETURN
```

Sample screen display.



How I wrote my first machine code program

Ian Turtle, of Ashby, Scunthorpe, explains how he got to grips with machine code, the problems he faced, and the triumph he achieved.



This is an account of how I wrote my first program in machine code, following the study of Toni Baker's book 'Mastering Machine Code on the ZX81'. It was a real struggle to get the program actually working, as many of my initial assumptions were proved wrong.

At first, I wrote a BASIC program to search through a program, and convert all the letters, numbers and the like to inverse video; this not only looks good, but allows REM statements to stand out in listings so long as there are not too many PRINT statements in the program.

The program sat at the bottom of program memory, and had to be typed in after the program which I wanted to transform had been entered. My BASIC program to do this is shown in Listing 1.

```
9500  FOR I=16509 TO PEEK 16396+256★PEEK
      16397
9510  IF PEEK I ≠ 234 THEN GOTO 9590
9520  LET I=I+1
9530  IF PEEK I=118 THEN GOTO 9590
9540  LET A= PEEK I
9550  IF A>63 THEN GOTO 9520
9560  LET A=A+128
9570  POKE I,A
9580  GOTO 9520
9590  NEXT I
9600  LIST
```

Listing 1

Line 9500 finds the end of the program area, and sets up a FOR. . .NEXT loop to search the listing. Line 9510 checks if the byte is a REM keyword. If not, it jumps to line 9590 and back around the loop. Once a REM statement is encountered, then it begins to execute the lines 9520 to 9580. Line 9520 increments the I count to the next byte in the REM statement. Line 9530 checks for a NEWLINE, and the end of the REM statement. If the end of the REM has been found, then the program again begins to search for another REM in memory. If the end has not yet been reached, then the code of the character is checked. If it is greater than 63 then it is likely to be a keyword with no inverse, or a letter/number that is already reversed, so the loop skips this byte.

If its value is less than, or equal to, 63, then 128 is added to the value to get the inverse (line 9560). Line 9570 then POKEs this value into memory and line 9580 returns to check the next byte in the program. Enter the program, and add a line like 9951 REM XXXXXXXX and RUN the program. The listing will appear with the Xs after the word REM in inverse as required.

Unfortunately, it takes the program three seconds to achieve this. Admittedly, three seconds is not long to wait, but if this program had to sort through a 15K program it would take minutes. To be frank, I thought this would be more trouble than it is worth.

Eureka!

The obvious answer was to put it into machine code. Unless you are a computer fanatic, then I would not lightly recommend you take the drastic step of deciding to tackle machine code.

If you find BASIC programming makes you tear your hair out in frustration with its error messages as you debug your latest masterpiece, do not attempt to write machine code, or you'll soon need the services of a hair-transplant specialist. It is the most frustrating experience I have ever gone through. Despite this, the result were well worth all the effort.

The frustration made me write a simple BASIC program to aid the development of machine code, but more about that later. Eventually, the program was written and debugged, and is shown in Listing 2.

	Op-code	Hex	Decimal	Bytes
START	LD HL,16549d	21A540	33,165,64	3
	LD DE,(16396)d	ED5B0C40	237,91,12,64	4
	INC HL	23	35	1
ENDCH	LD A,D	7A	122	1
	CP H	BC	188	1
	JRNZ REMCH	2003	32,3	2
	LD A,E	7B	123	1
	CP L	BD	189	1
	RET Z	C8	200	1
REMCH	LD A(HL)	7E	126	1
	CP 234d	FEEA	254,234	2
	JRNZ ENDCH	20F3	32,243	2
INREM	INC HL	23	35	1
	LD A,(HL)	7E	126	1
	CP118	FE76	254,118	2
	JR Z ENDCH	28ED	40, 237	2
	CP 64	FE40	254,64	2
	JRNC INREM	30F6	48,246	2
	ADD A,128	C680	198,128	2
	LD (HL),A	77	119	1
	JR INREM	18F1	24,241	2

Listing 2



Programming in machine code can make you old before your time.

This is simply a straight translation of the BASIC into machine code. Location 16514 inwards (ie a REM statement as the first line of a ZX81 program) is my favourite place to store machine code, so the program was written to occupy these memory locations. In theory, however, the program is completely relocatable, as there are no JPs as opposed to JRs (things you have to come to terms with when programming in machine code). In practice, this assumption proved incorrect.

The HL register at the start of the listing is set to 16549 instead of 16509 to skip over the actual REM statement. If this was not done, the machine code would search itself, find the 243d one instruction after REMCH and assume this was a REM statement. Then it would alter all the machine code which followed, until it reached the 118 two instructions after INREM. Obviously, this would cause a crash.

If, then, the machine code was relocated, it would still begin its search at memory location 16549, and miss any early REM statement. The solution was simply to substitute 16509 in START.

The idea behind relocating the routine was to place it above RAMTOP, to save having to reload machine code over and over again in a programming session. However, disaster struck when the machine code was used to invert all the REM statements in the actual 'machine code loader' program. For some reason, it corrupted the line which read:

3050 IF B$ = "D" THEN LET A$ = STR$ B

To find out why, it was necessary to find out how the 8K ROM stored line numbers.

I discovered that the computer stores a line number in four bytes. The first two bytes contain the value of the line number, and the second two bytes indicate how many bytes long the line is. So a line 10, LET A PEEK 1, would have the four 'line number' digits reading 00 0A 06 00. Strangely, the computer has a 'double standard' here. The line number is stored with the high byte followed by the low byte. The length of the line, however, is stored the other way round, with the low byte first.

Back to the beginning

To return to the original problem, the line number 3050 will be stored in memory as 0BEA, where EA is the value the program searches for. So, the length of line marker will be overwritten – not a good way to keep a bug-free program. With this knowledge, it is obvious that there is a set of numbers that would cause this problem. It was luck that uncovered this bug (the numbers are 234, 490, 746, 1002, 1258, 1514, 1770 . . .).

This was all very satisfying in its way, but it meant the original machine code program was practically useless, as it would have meant the user had to search through the listing and change every occurrance of the line number set to something else, running the code, then changing them all back. A return to the BASIC listing would obviously be a better idea. "Back to the drawing board", I thought, and attempted a new approach.

I do not think it is just me. However, I find it very difficult to produce a brand new method, or idea, straight after an old one had been tested and made to work. However, the papers with a disassembled listing (BASIC) were close to hand. It was while rereading these that the idea of using the length of line indicator occured. Obviously, I cannot claim monopoly on this idea – I had read of its use elsewhere, but I had not previously thought of using it here. Anyway, it sounded rather complex. In fact, it did cause the machine code to be a little more complicated, but not too difficult.

The machine code listing was longer, but it used the fact that the REM indicates the end of a listing by two Newlines on the trot to check for the end of the program. It was with this listing I was finally satisfied. HL was loaded at START with 16509 since I had finally decided to go for storing the machine code above RAMTOP for more convenience. This was done as follows:

```
ENTER POKE 16388, 216
      POKE 16389, 127
      NEW
```

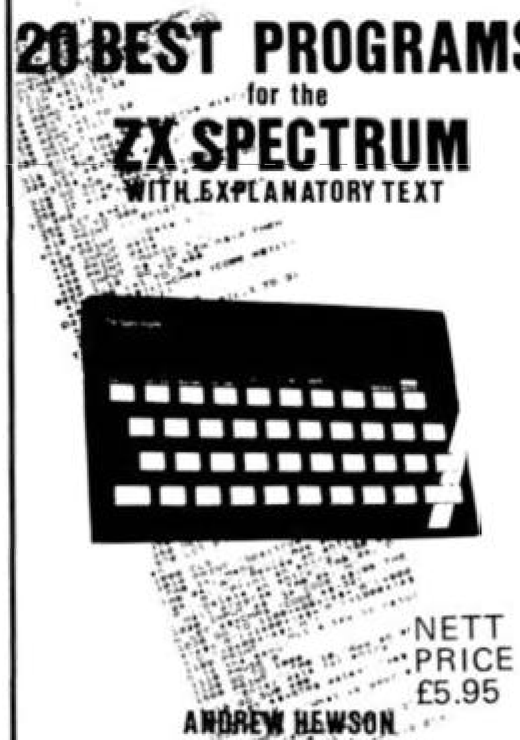
Then the machine code can be written, starting at address 32729.

So, after many hours , the listing below was produced, which did the job demanded of it in the blink of an eye,as opposed to minutes. I tested it on the longest program I could find, and it took no noticeable running time at all. As a conclusion, I'd say that if you're willing to stick at programming in machine code, the results will be well worth the cost in frustration.

	Op-code	Hex	Decimal	Bytes
START	LD HL,16509d	217D40	33,125,64	3
LSKIP	INC HL	23	35	1
	INC HL	23	35	1
	LD E,(HL)	5E	94	1
	INC HL	23	35	1
	LD D,(HL)	56	86	1
	INC HL	23	35	1
	LD A(HL)	7E	126	1
	CP 234d	FEEA	254,234	2
	JRZ ENDCH	2807	40,7	2
	ADD HL,DE	19	25	1
	LDA,(HL)	7E	126	1
	CP 118d	FE76	254,118	2
	RET Z	C8	200	1
	JR LSKIP	18EE	24,238	2
ENDCH	INC HL	23	35	1
	LD A,(HL)	7E	126	1
	CP 118d	FF76	254,118	2
	JRZ CONTS	2809	40,9	2
	CP 64d	FE40	254,64	2
	JRNC ENDCH	30F6	48,246	2
	ADD A,128	C680	198,128	2
	LD (HL), A	77	119	1
	JR ENDCH	18F1	29,241	2
CONTS	INC HL	23	35	1
	JR LSKIP	18DC	24,220	2

Listing 3

ANDREW HEWSON'S BOOKS FOR THE SPECTRUM



Mr HELPLINE – the man who answers your ZX queries in his column in Sinclair User, the author of HINTS & TIPS FOR THE ZX80 and HINTS & TIPS FOR THE ZX81 now presents:

* 20 original programs for you to load into your Spectrum.
* 20 interesting programs for you to enjoy and learn from.
* 20 great programs to teach you about fixed and variable length records, binary searches, bubble sorts, graphic displays and much, much more . . .

Program titles include:

Machine Code Editor – Write, modify, extend and load machine code using this all-basic program. No need to use an Assembler when you have this program.

Index File – Learn about fixed length records, save numeric and string information, add to, sort, modify, delete and print your records. Ideal as a computer based filing system.

Duckshoot – Learn how to manipulate the attributes file *and* have fun at the same time.

Graphix – Construct up to 210 graphics characters with the full on-screen editing facilities, enhance and modify them and recall them later to build a detailed display to save on cassette.

Spiromania – A program to stretch your artistic talents, imagination and ingenuity. Draws a limitless variety of curves and spirals.

Plus: FOOTBALL, DIGITISER, DIARY and many more.

Available through Computer Bookshops and W.H. SMITH.

40 BEST MACHINE CODE ROUTINES FOR THE ZX SPECTRUM £5.95

by Andrew Hewson and John Hardman.

Section A: Three chapters explaining what you need to know about Z80 machine code on the Spectrum.

* How to load and save machine code.
* How to use the system variables.
* How memory is organised.
* How program lines are stored.
* How to use the stack, the display, the attribute files.
* How to call ROM routines – where they are and what they do.
* The structure of Z80 code – plus a valuable glossary.

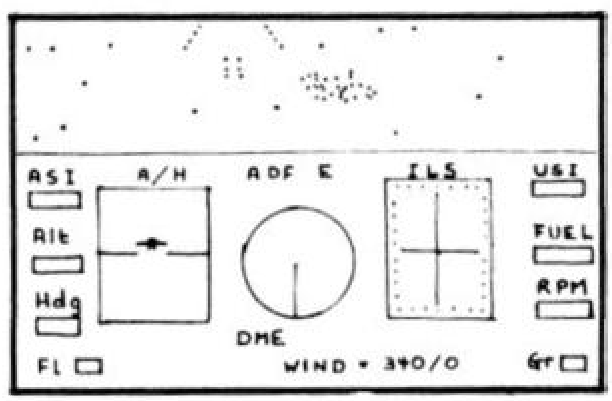
Section B: 40 routines including,

* Scroll – up, down, side to side by pixel or by character.
* Search and replace, token swap, string search.
* Rotate character, invert character – horizontally and vertically.
* Line remember – including GOSUBs, GOTOs, RUN etc.

All routines are relocatable (except 'line remember')

This book teaches the beginner all he needs to know in a simple, easy-to-learn form and its an invaluable reference work for the expert too!

NIGHTFLITE FOR ZX SPECTRUM £5.95



Fly your own aircraft from take off to landing via navigational beacons, over mountains and using a fully detailed direction finding and instrument landing system. *You are the Pilot of a light aircraft flying at night. *You must use your skill and judgment to fly your aircraft accurately over radio beacons and then land safely on the runway *Hazards are mountains and cross winds *Instruments: Artificial Horizon, Non Directional Beacon, VHF Omnidirectional Range, Instrument Landing System *Readouts: Gear, Flap, Air Speed, Distance Measuring Equipment, Vertical Speed, RPM and heading *Visual display of runway on approach *5 Modes from Take off to Autopilot *Happy landings *

Z80 OP CODES £1.45

A must for the beginner and the experienced programmer alike. This hand ready reckoner lists all 600 plus Z80 machine code instructions in decimal and hexadecimal with their mneumonics. Each Op Code is succinctly explained and cross-referenced. Supplied in a protective transparent wallet for easy reference and durability.

PILOT ZX81 £5.95

Essentially the same as Nightflite but without the hi res graphics.

PUCKMAN ZX81 £5.95

All action display. Best score to date, scour the maze for food, dodge the ghost, eat a strawberry and attack the ghosts.

HINTS & TIPS FOR THE ZX81 £3.95

Available by direct mail order or through computer bookshops and W.H. Smith.

THE PROGRAMS

The programs advertised here are available from computer bookshops with a software stand, and many independent micro-computer shops.

RETAILERS

We are continually on the look out for new outlets. If you are interested in stocking the items advertised here, write to us for our trade rates.

URGENT

We require high quality Spectrum and Dragon software. Good royalties paid. Send your samples today for fast evaluation.

QUANTITY	PRODUCT	COST
	TOTAL	

MAKE CHEQUES/PO's PAYABLE TO: HEWSON CONSULTANTS

NAME ....
(block capitals please)
ADDRESS ....

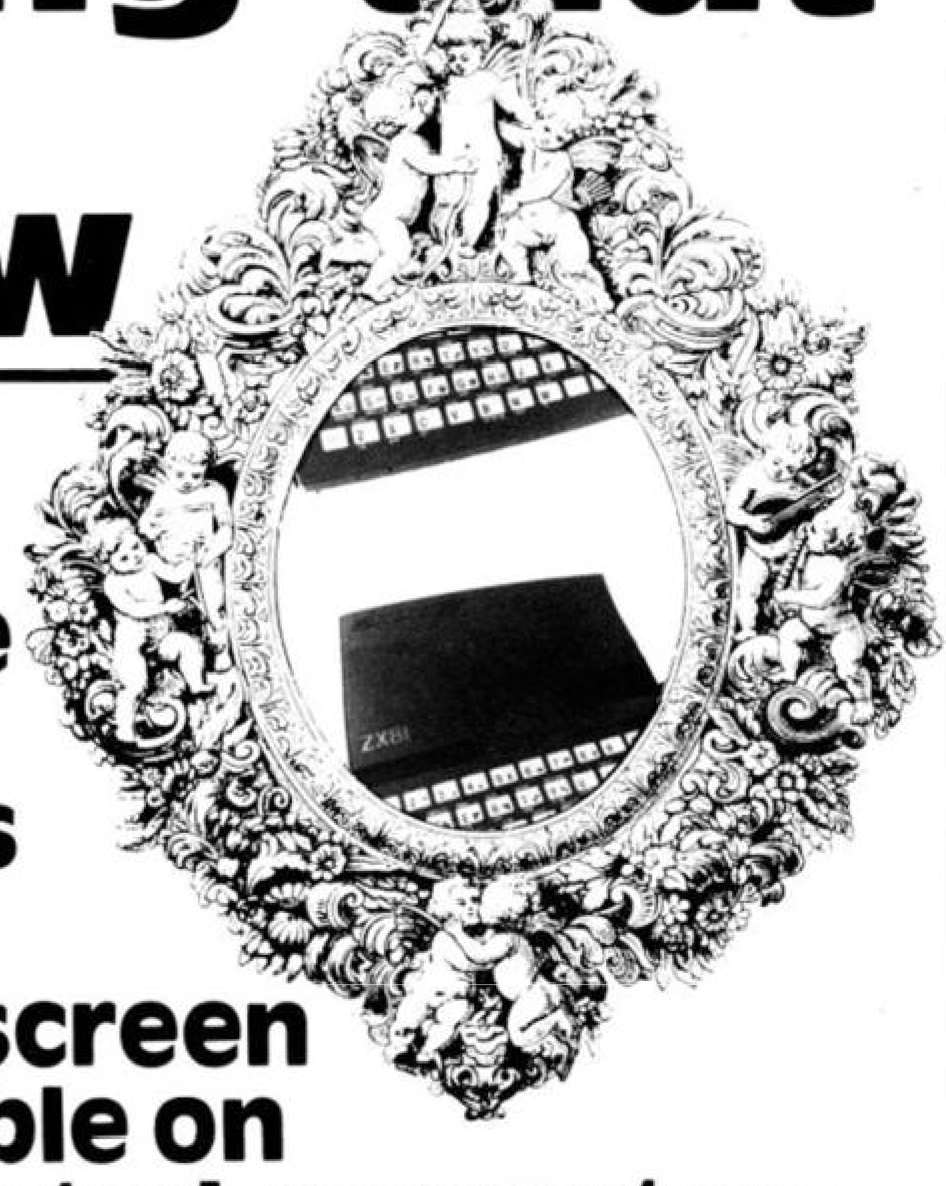
....

My Access/Barclaycard No. is ....

Signed ....

Post to: **HEWSON CONSULTANTS, DEPT SU 60A St MARY'S STREET, WALLINGFORD, OXON OX10 0EL. TEL (0491) 36307.**

Scrolling that screen window



J Elliott from Gloucester describes some machine code routines he has developed to emulate some screen controls available on more sophisticated computers.

As a teacher I have access to both an RML 380Z and 480Z in the course of teaching Computer Studies. These computers allow fairly easy screen control. In particular they allow the definition of "screen windows" at any point on the screen. These windows may be scrolled independently of the remainder of the screen. In addition to this, the "GRAPH" command restricts scrolling to the bottom four lines of the screen.

My own ZX 81 does not have these facilities. The "PRINT AT" command is the nearest equivalent. I decided to attempt to emmulate these facilities with machine code subroutines. The three routines to be described increase the screen handling abilities of the ZX81 quite considerably.

The first routine defines a rectangular window of any size at any point on the screen. Once defined, the window can be instantly filled with any character, or cleared, simply by calling the routine. Any of these parameters can be changed at any time from the master program. This means that several different windows can be used with the same program.

The second routine uses the same principle as the first. A window is again defined — any size and any position on the screen. When the routine is called, the contents of the window are scrolled up through one line and the last line is cleared. The routine can be set to fill the last line with any character instead of clearing it.

The last routine is a very simple screen reverse. Every character on the screen is removed and replaced with its inverse. This is very effective for explosions.

Using the routines

The easiest way of using the routine is as follows;

First type in listing 4 and save at least twice.

Save 114 bytes of high memory by poking addresses 16388 and 16389 with a new value for ramtop. With a 16K machine, suitable values are ; POKE 16388,140 POKE 16389,127.

If you now enter 'NEW' you will have the space above 32652 free for the machine code.

Now reload listing 4 and 'RUN'. The code will be loaded to the area of memory you enter when prompted by the program. You will now be prompted to enter size and positions for your window or scroll window and also fill character. Having done this you are offered the option of redefining, testing or ending the program. When you end the program is automatically 'NEWed' so that you can easily enter the BASIC code which will call the routines.

To call the routines, the following commands can be inserted at any point in the BASIC;

i) Fill or clear window — LET L=USR START

ii) Scroll one line of window — LET L=USR (START+39)

iii) Inverse screen-LET L=USR (START+94)

Where START is the start address of the machine code which must have been previously loaded the parameters of the windows can be changed easily from

within the BASIC program using the routines, by 'POKEing' the following addresses with suitable values;

Fill window

(START +5) = last line of window
(START +7) = number of lines of window
(START +20) = 32-start column of window
(START +25) = columns from end of window to line end
(START +30) = fill character code

Scroll window

(START +44) = last line number in window
(START +46) = number of lines in window
(START +59) = 32-start column number
(START +64) = columns from end of window to line end
(START +85) = scroll character code

The program shown in listing 5 demonstrates how this is done. It produces random windows and fills or scrolls them with random characters. It can produce some quite effective patterns!

I have included listing 1, 2 and 3 for those of you interested in the assembler for these routines.

Memory

To run the routines as described you need at least 3.5K of memory. Those of you with less than this will need to create a display file by 'PRINTing' spaces to the screen before you attempt to use the routines. The routines will work perfectly well with an 8K ROM ZX80 but the following changes must be made to the loading and demonstration programs;

Loading program

Line 100 INPUT D$
Line 520 INPUT M$

Demonstration program

Add line 115 PAUSE 100
Add line 116 POKE 16437,255

Whenever these routines are used, you will have to use 'PRINT AT' to print to the windows. The normal 'PRINT' statement will print in the normal way to the whole screen.

```
2A 0C 40   START      Ld HL, (Display File)  ;find screen in memory
23                    INC HL                 ;find first screen location
0E x                  Ld C,x                 ;number of lines to rectangle
3E x       RECTANGLE  Ld A,x                 ;rectangle length end
B9                    CPC                    ;start reached?
30 06                 JRNC BEGIN             ;
11 21 00              Ld DE 0021H            ;
19                    ADD HL, DE             ;move one line down screen
18 12                 JR END                 ;
06 20      BEGIN      Ld B 20H               ;length of full line
3E x       COLUMN     Ld A,x                 ;(32-start column number)
B8                    CPB                    ;start of rectangle reached?
38 07                 JRC NEXT               ;
3E x                  Ld A,x                 ;number of columns to clear
B8                    CPB                    ;not finished?
30 02                 JRNC NEXT              ;
36 x                  Ld (HL),x              ;put character to screen
23         NEXT       INC HL                 ;next screen position
10 F1                 DJNZ                   ;repeat to end of line
23                    INC HL                 ;step past line end
0D         END        DEC C                  ;
20 E0                 JRNZ RECTANGLE         ;repeat to end of screen
C9                    RET                    ;
```

Listing 2: Screen scroll

```
2A 0C 40   START      Ld HL, (Display File)  ;find screen in memory
23                    INC HL                 ;find first screen location
0E x                  Ld C,x                 ;lines to end of window
3E x       SCROLL     Ld A,x                 ;lines to scroll
B9                    CPC                    ;start reached?
30 06                 JRNC BEGIN             ;
11 21 00              Ld DE 00 21H           ;
19                    ADD HL, DE             ;move down one line
18 22                 JR END                 ;
06 20      BEGIN      Ld, B,20H              ;length of full line
3E x       COLUMN     Ld A,x                 ;(32-start column number)
B8                    CPB                    ;start of window?
38 17                 JRC NEXT               ;
3E x                  Ld A,x                 ;number of columns to scroll
B8                    CPB                    ;not finished?
30 12                 JRNC NEXT              ;
3E 01                 Ld A, 01H              ;
B9                    CPC                    ;last line?
28 0B                 JRZ LAST               ;
E5                    PUSH HL                ;save screen position
11 21 00              Ld DE 00 21H           ;
19                    ADD HL, DE             ;move down one line
D1                    POP DE                 ;get screen in DE
7E                    Ld A(HL)               ;move character up one line
12                    Ld (DE)A               ;
EB                    EX DE HL               ;get screen in HL
18 02                 JR NEXT                ;
36 00      LAST       Ld(HL),0               ;clear last line
23         NEXT       INC HL                 ;get next screen position
10 E1                 DJNZ COLUMN            ;repeat if not finished
23                    INC HL                 ;step past line end
0D         END        DEC C                  ;
20 D0                 JRNZ SCROLL            ;scroll finished?
C9                    RET                    ;
```

Listing 3: Screen reverse

```
2A 0C 40   START      Ld HL, (Display file)  ;find screen in memory
23                    INC HL                 ;find first screen location
0E 16                 Ld C,16H               ;lines in screen
06 20      LINE       Ld B,20H               ;columns on line
7E         NEXT       Ld A,(HL)              ;reverse character at current
C6 80                 ADD A 128              ;screen position
77                    Ld (HL),A              ;
23                    INC HL                 ;get next screen position
10 F9                 DJNZ NEXT              ;repeat if line not finished
23                    INC HL                 ;step past line end
0D                    DEC C                  ;
20 F3                 JRNZ LINE              ;repeat if screen not finished
C9                    RET                    ;
```



Listing 4: Basic loader

```
10 REM LOADING PROGRAM FOR SCREEN ROUTINES
20 LET A$ = "2A0C40230E123E0AB930061121001918
   1206203E18B838073E08B8300236802310F1230D
   20E0C92A0C40230E123E0AB9300611210019182 2
   06203E18B838173E08B830123E01B9280BE51121
   0019D17E12EB180236002310E1230D20D0C92A0C
   40230E1606207EC680772310F9230D20F3C9"
30 PRINT "ENTER START ADDRESS FOR CODE"
40 INPUT B
50 FOR C=B TO B+113
60 POKE C. (16*CODE A$)+CODE A$(2)-476
70 LET A$=A$(3 TO)
80 NEXT C
85 CLS
90 PRINT "ENTER I TO DEFINE WINDOW, 2 TO DEFINE
   SCROLL OR 3 TO END"
100 LET D$= INKEY$
110 IF D$<"I"OR D$ >"3" THEN GOTO 100
120 IF D$="3" THEN GOTO 500
130 IF D$="1" THEN LET J$="DEFINE WINDOW"
140 IF D$="2" THEN LET J$="DEFINE SCROLL"
145 CLS
150 PRINT AT 2,9;J$
160 PRINT AT 4,0; "ENTER START LINE"
165 INPUT E
170 PRINT AT 4,0; "ENTER FINISH LINE"
175 INPUT F
180 PRINT AT 4,0; "ENTER START COLUMN"
185 INPUT G
190 PRINT AT 4,0; "ENTER FINISH COLUMN"
195 INPUT H
200 PRINT AT 4,0; "ENTER FILL/SCROLL CHARACTER"
205 INPUT K$
210 IF (((CODE K$ >63) AND (CODE K$ <128)) OR (CODE
    K$ >191)) THEN GOTO 205
220 LET L=0
230 CLS
240 IF D$="2" THEN GOSUB 400
250 POKE B+5, F
260 POKE B+7, F-E
270 POKE B+20, 32-G
280 POKE B+25, 32-(G+H)
290 POKE B+30-L, CODE K$
300 GOTO 85
400 LET B=B+39
410 LET L=16
420 RETURN
500 CLS
510 PRINT "ENTER 1 TO TEST YOUR WINDOW, 2 TO
    REDEFINE OR 3 TO NEW THE PROGRAM"
520 LET M$=INKEY$
530 IF M$ <"1" OR M$ >"3" THEN GOTO 520
540 IF M$"3" THEN NEW
550 IF M$="2" THEN GOTO 85
560 LET N=USR B
570 FOR P=1 TO (F-E)
580 LET N=USR (B+39)
590 NEXT P
600 PAUSE 100
610 POKE 16437,255
620 GOTO 500
```

Listing 5: Demonstration routine

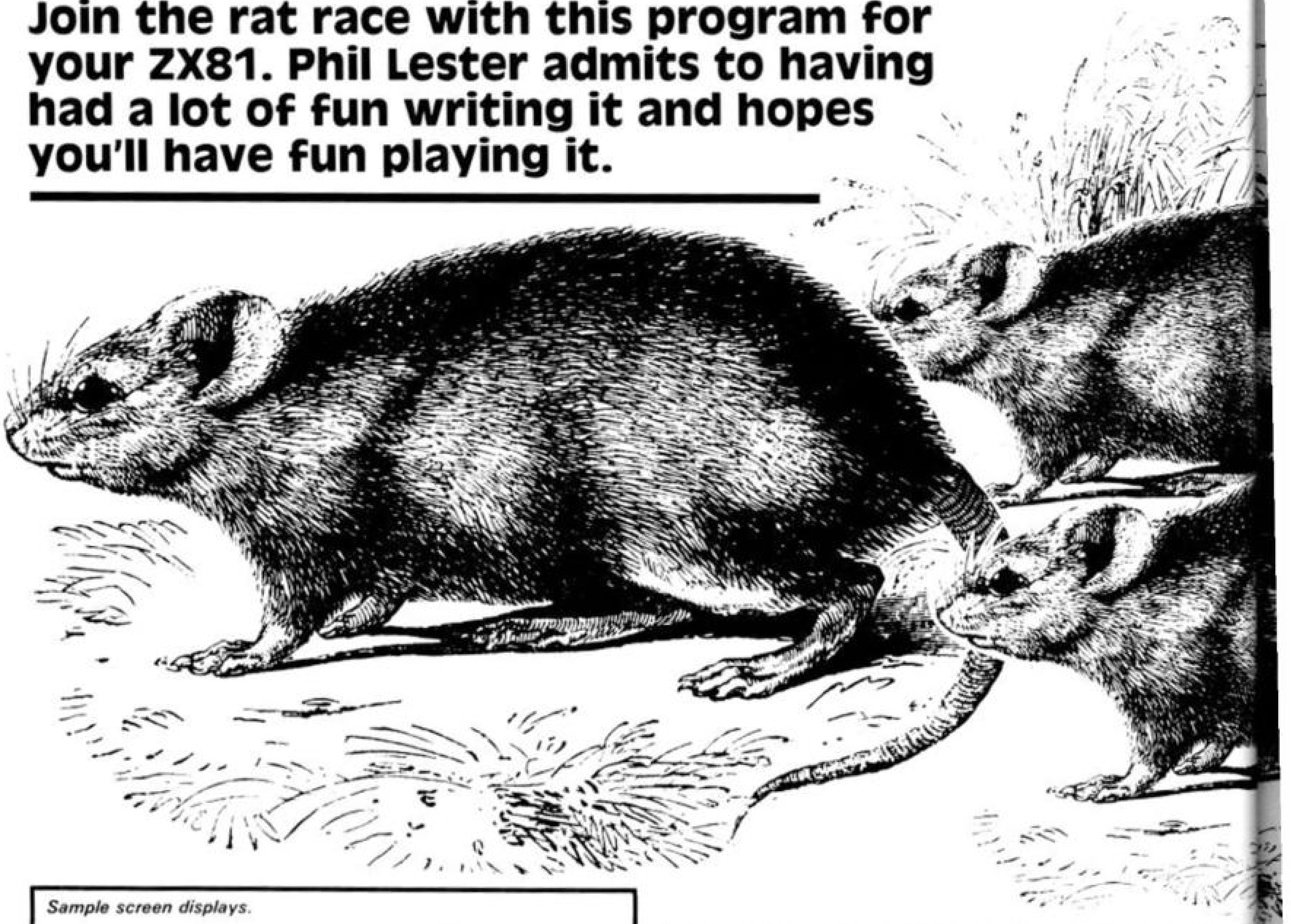
```
10 REM SCREEN ROUTINES MUST BE IN MEMORY
20 PRINT "ENTER START ADDRESS OF MC ROUTINES"
30 INPUT A
40 CLS
50 LET B= INT(RND*22)+1
60 LET C= INT(RND*B)+1
70 LET D= INT(RND*32)+1
80 LET E= INT(RND*(32-D))
90 LET F= INT(RND*63)+(128*((RND*2)<1))
100 LET G= INT(RND*3)+2
110 GOSUB (100*G)
120 GOTO 50
200 POKE (A+5),B
210 POKE (A+7),C
220 POKE (A+20),32-D
230 POKE (A+25),E
240 POKE (A+30),F
250 LET H= USR A
260 RETURN
300 POKE (A+44),B
310 POKE (A+46),C
320 POKE (A+59),32-D
330 POKE (A+64),E
340 POKE (A+85),F
350 FOR H=1 TO C
360 LET J= USR (A+39)
370 NEXT H
380 RETURN
400 LET H= USR (A+94)
410 RETURN
```

Rat race

Join the rat race with this program for your ZX81. Phil Lester admits to having had a lot of fun writing it and hopes you'll have fun playing it.

Rat Race is a game for two to four players and occupies 4K of memory, so you'll need your 16K RAM Pack.

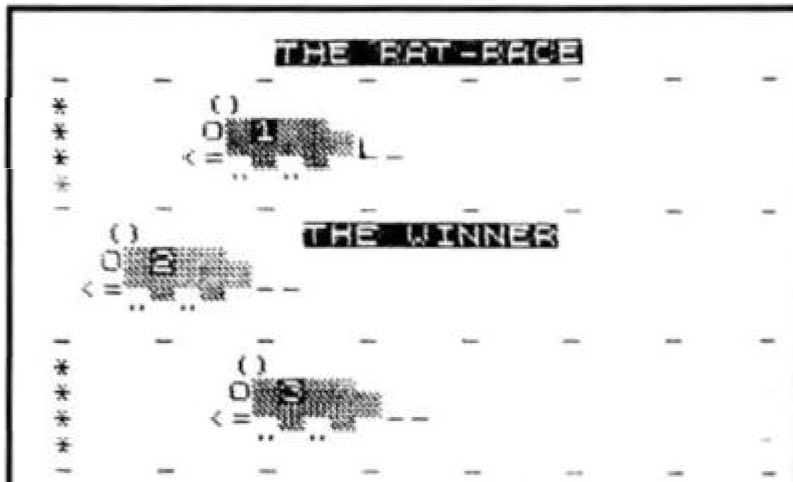
Each player has a rat each and £200 with which to have a bet (you may only bet on your rat). If your rat wins then you win the amount you wagered and this is added to your cash total. Should you lose, the amount of your flutter is deducted from your cash total. Should your cash total reach zero, you are pronounced broke



Sample screen displays.

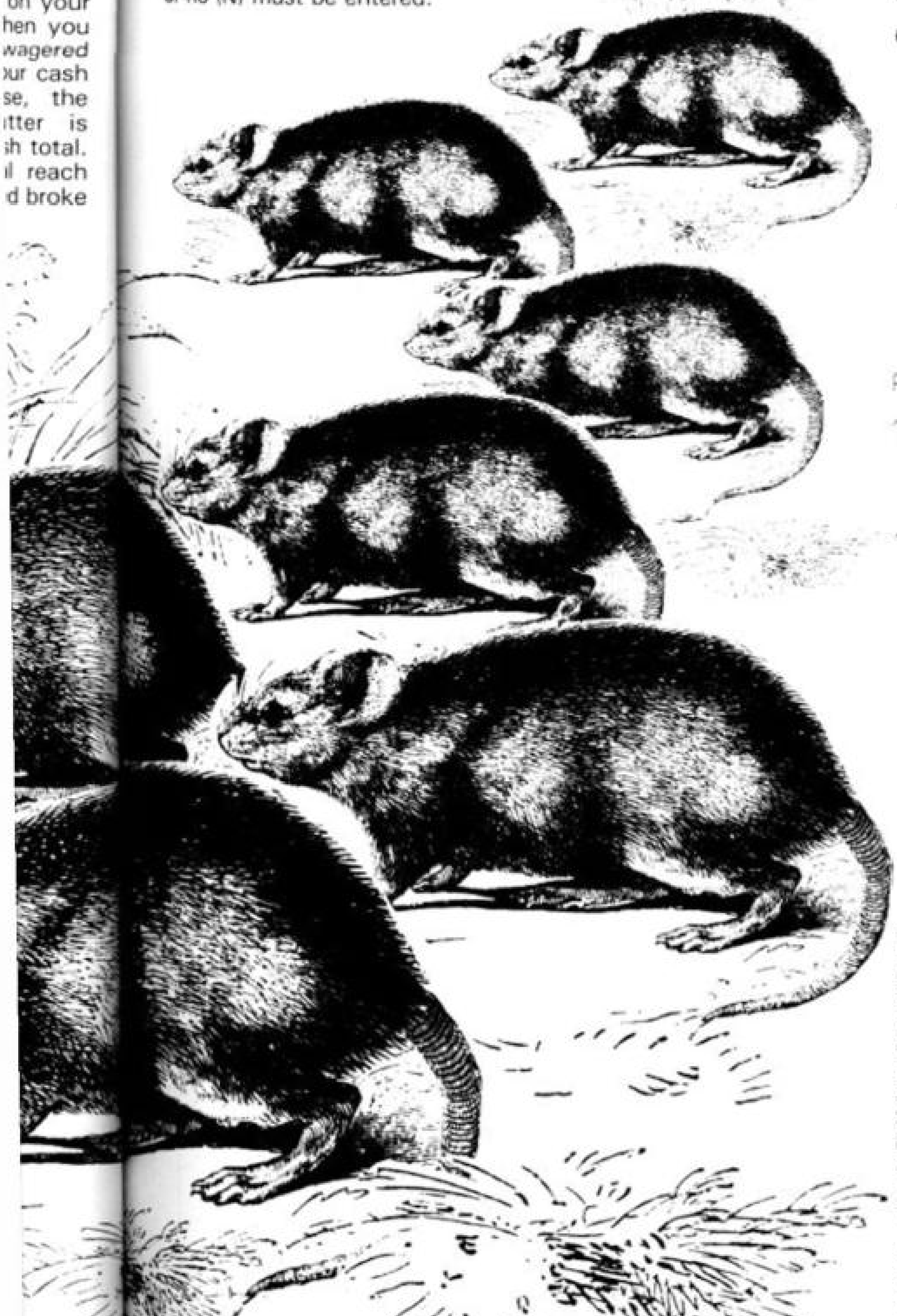
```
PLAYER NO.1 NAME? LOUIE
PLAYER NO.2 NAME? DOUIE
PLAYER NO.3 NAME?
```

```
LOUIE       YOU HAVE £200
WHAT WILL YOU BET ON NO.1? £100
HOUIE       YOU HAVE £200
WHAT WILL YOU BET ON NO.2? £150
DOUIE       YOU HAVE £200
WHAT WILL YOU BET ON NO.3? £50
```



and out of the game.

Once the amount of each bet has been recorded, the race is on. Once all players have run out of money, another game is offered to which the reply yes (Y) or no (N) must be entered.



```
75 CLS
80 DIM A(4)
85 DIM B(4)
90 DIM R(4)
95 DIM A$(4,8)
100 PRINT AT 10,5;"HOW MANY PLA
YERS? (2-4)"
105 INPUT P
108 IF P>4 OR P<2 THEN GOTO 105
110 CLS
200 FOR N=1 TO P
205 PRINT ,,,,"PLAYER NO.";N;"
NAME? ";
210 INPUT A$(N)
220 PRINT A$(N)
230 NEXT N
235 GOSUB 4000
250 FOR N=1 TO P
256 LET R(N)=200
257 NEXT N
290 CLS
295 LET U=0
300 FOR N=1 TO P
303 IF R(N)=0 THEN GOTO 350
305 PRINT ,,,,A$(N);" YOU HAVE
£";R(N)
306 PRINT ,,"WHAT WILL YOU BET
ON NO.";N;"?";
310 INPUT B(N)
312 IF B(N)>R(N) THEN GOTO 310
314 PRINT " £";B(N)
320 NEXT N
330 GOTO 690
350 PRINT ,,,,A$(N);" IS BROKE.
.."
355 LET U=U+1
356 IF U=P THEN GOTO 400
360 NEXT N
370 GOTO 690
400 GOSUB 4000
405 CLS
410 PRINT AT 8,5;"YOU ARE ALL B
ROKE"
415 PRINT AT 12,5;"ANOTHER GAME
? Y/N"
420 INPUT I$
425 IF I$="Y" THEN GOTO 435
428 GOTO 5000
435 PRINT AT 16,5;"SAME PLAYERS
? Y/N"
440 INPUT I$
445 IF I$="Y" THEN GOTO 250
450 GOTO 75
690 GOSUB 4000
700 CLS
705 FOR L=2 TO P*5
710 PRINT AT L,0;"*"
720 NEXT L
750 FOR L=1 TO P*6 STEP 5
760 PRINT AT L,0;"-    -    -    -
-    -    -    -"
770 NEXT L
780 PRINT AT 0,9;"THE RAT-RACE"
840 FOR N=1 TO P
845 LET A(N)=20
850 NEXT N
1000 FOR N=1 TO P
1010 LET L=N*5-3
1020 LET A(N)=A(N)-SGN (RND*3-.5
)-(A(N)>21)
1040 GOSUB 2000
1050 NEXT N
1060 GOTO 1000
2000 PRINT AT L,A(N);"   ()  "
2010 PRINT AT L+1,A(N);"   O ";CH
R$ (N+156);"     "
2020 PRINT AT L+2,A(N);"  <=
--"
2030 PRINT AT L+3,A(N);"     "" "" "" ""
"
2050 IF A(N)=0 THEN GOTO 2200
2100 RETURN
2200 FOR N=0 TO 4
2205 PRINT AT L,10;"          "
2210 FOR Z=0 TO 6
2211 NEXT Z
2220 PRINT AT L,10;"THE WINNER"
2230 FOR Z=0 TO 6
2231 NEXT Z
2235 NEXT N
3000 FOR N=1 TO P
3010 IF R(N)=0 THEN GOTO 3060
3020 IF A(N)=0 THEN GOTO 3050
3030 LET R(N)=R(N)-B(N)
3040 NEXT N
3045 GOTO 290
3050 LET R(N)=R(N)+B(N)
3060 NEXT N
3065 GOTO 290
4000 FOR Z=0 TO 50
4010 NEXT Z
4020 RETURN
5000 PRINT ,,,,,,,"        OKAY BYE"
```



MEMOPAK 16K For those just setting out on the road to real computing, this pack transforms the ZX81 from a toy to a powerful computer. Data storage, extended programming and complex displays become feasible.
For even greater capacity, memory packs can be added together (16 + 16 + 16K or 16 + 32K). The MEMOPAK 32K and the MEMOPAK 64K offer large memories at economical prices.

MEMOTECH

The Memotech approach to microcomputing is to take the well-prove and popular ZX81 as the heart of a modular system. This small computer houses the powerful Z80A processing unit and acts as the central processor module through which the MEMOPAKS operate.

Memotech has a reputation for professional quality, producing units which are designed to fit perfectly, to look well-balanced, and to work efficiently and reliably.

The modular approach gives ZX81 owners the freedom to design the system they really need. Furthermore, the intercompatibility of the modules ensures that later additions will click straight in, to give you a system that grows with your ambitions and abilities.

As one example, a system with 16K of memory and MEMOCALC is all that is required to perform sophisticated numerical calculations giving the same results as a computer at 10 times the price. The problem may be as complicated as a cash flow or production schedule, or as simple as household accounts or pocket money budgeting. If the bank manager wants to see the cash flow, then a single print instruction to the Centronics I/F will give a printout which is more than acceptable to any bank.

The example system which is shown, on the other hand, would satisfy the needs of someone who wanted to enter data via a light-touch keyboard, construct and label graphs, and then copy the screen to an 80-column printer. Only 16K of memory is used here but with additional memory, more than one video page can be stored. Up to 7 successive pages can be displayed cyclicly to give animated displays.

MEMOCALC The screen display behaves as a 'window' on a large sheet of paper on which a table of numbers is laid out. The maximum size of the table is determined by the memory capacity, and with a MEMOPAK 64K a table of up to 7000 numbers with up to 250 rows or 99 columns can be specified. Each location in the table can be either a number which is keyed in or a formula which generates a number. Every time the command to 'calculate' is given, all the formulae in the table are re-evaluated. Spreadsheet analysis started as an aid to cash-flow analysis, but this powerful tool has now been generalised and MEMOCALC with its special ability to perform iterative calculations is invaluable in the performance of numerical tasks.



16K	£26.00	+ £3.90	VAT	£29.9
32K	£43.43	+ £6.52	VAT	£49.9
64K	£68.70	+ £10.30	VAT	£79.0
HRG	£34.70	+ £5.20	VAT	£39.9
CI/F	£34.70	+ £5.20	VAT	£39.9
MEMOCALC	£26.00	+ £3.90	VAT	£29.9
Z80 ASSEMBLER	£26.00	+ £3.90	VAT	£29.9
KEYBOARD WITH BUFFER	£43.43	+ £6.52	VAT	£49.9

Memotech products are available at larger branches of WHSMITH

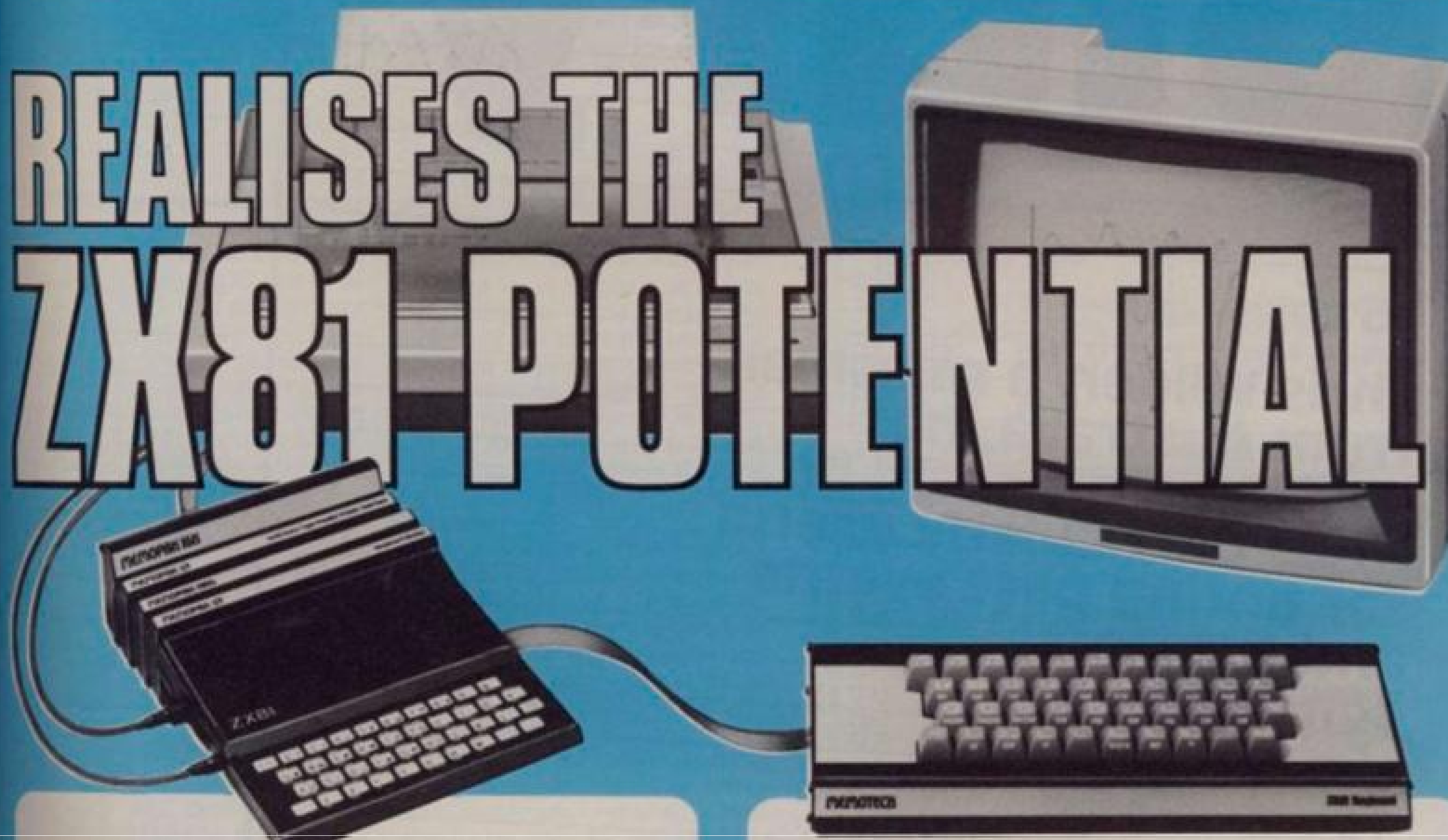


MEMOPAK HRG This pack breaks down the constraints imposed by operating at the ZX81 character level and allows high definition displays to be generated. All 248 × 192 individual pixels can be controlled using simple commands, and the built in software enables the user to work interactively at the dot, line, character, block and page levels. Scrolling, flashing and animation are all here.



MEMOPAK Centronics I/F The BASIC commands LPRINT, LLIST and COPY are used to print on any CENTRONICS type printer. All ASCII characters are generated and translation takes place automatically within the pack. Reverse capitals give lower case. Additional facilities allow high resolution printing. The full capabilities of your printer are now under the control of the ZX81.

REALISES THE ZX81 POTENTIAL



MEMOPAK Z80 Assembler This click-in EPROM based pack accepts standard Z80 assembly language mnemonics to allow you to write faster and more compact programs. It has its own ADD, EDIT, LIST, ASSM and QUIT functions, the editor allowing insertion, deletion, automatic line renumbering and error checking. Source code and object code listings can be displayed and printed in decimal or hex format.

MEMOTECH Keyboard The light-touch positive stop keys of this elegant typewriter-pitch keyboard allow you to work faster, more accurately and more confidently. To speed you along we have added an extra SHIFT key to the array at top right. The keyboard is attached by a cable to the Keyboard Buffer which fits in amongst your other Memopaks or straight onto the back of your ZX81.

To ensure that your expectations are realised, care is taken at every stage to design features into the system to anticipate your frustrations and to forestall them. For example:

A) Memories are cumulative e.g. 16K and 32K can be added to the MEMOPAK 16K or even to the Sinclair 16K RAM pack.

B) The HRG firmware allows commonly used constructions (such as scrolling, shading and labelling graphs), which might otherwise be beyond the user's programming capabilities, to be evoked by a few simple commands.

C) The Centronics I/F converts ZX81 character codes into ASCII and extends the print line to the width of the printer, still using the LLIST, LPRINT and COPY commands.

Looking forward, Memotech will continue to back the ZX81 through 1983 with fast storage devices, pressure sensitive electronic drawing boards and more software packs including a wordprocessor and an RS232 interface.

MEMOPAKS may be ordered by post (cheque, Access/Barclaycard quoting number) or by telephone. Please make cheques payable to Memotech Ltd. and please include £2.00 per unit for packaging and postage inland (overseas £3.00).

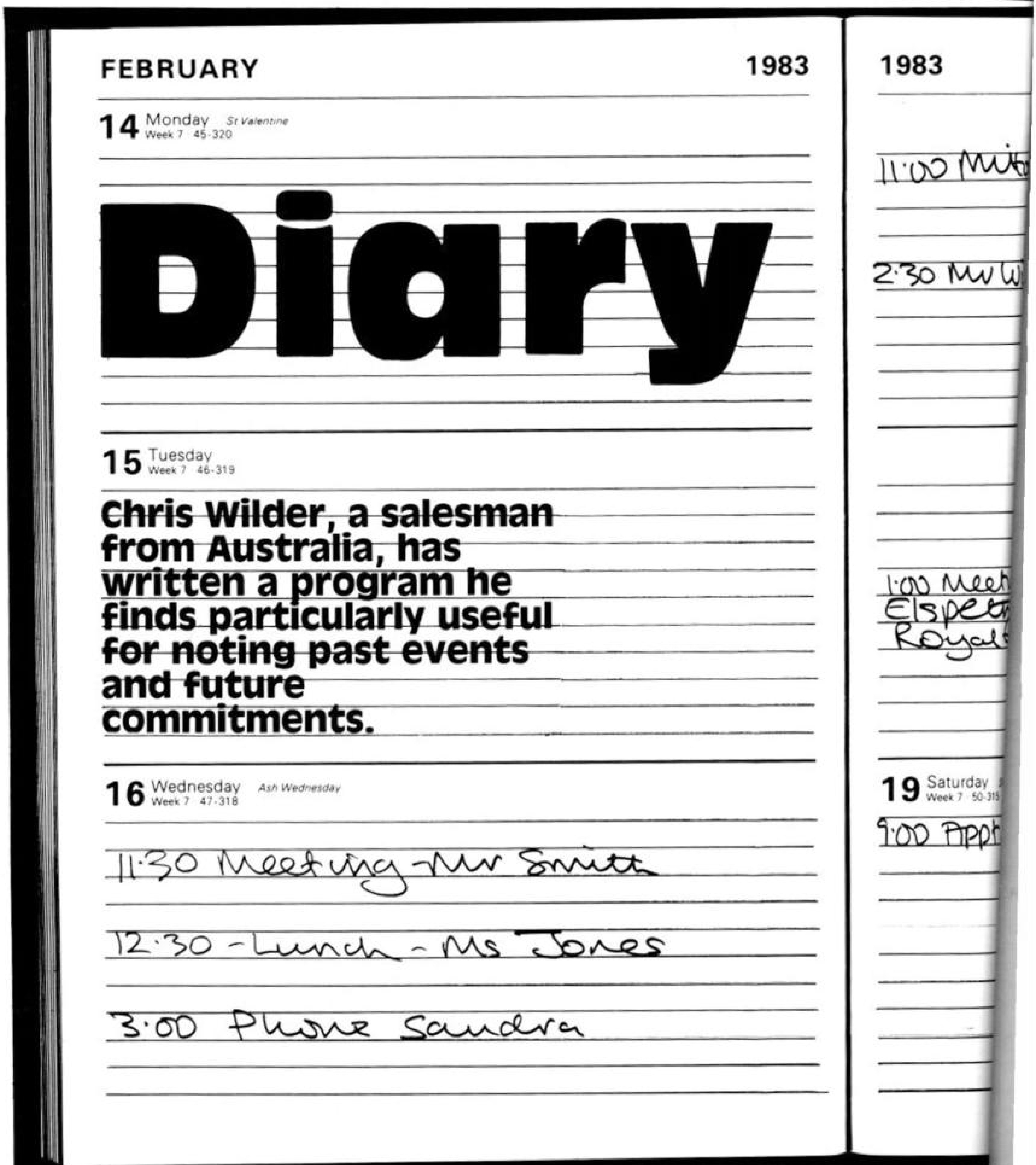
We want to be sure you are satisfied with your Memopak – so we offer a 14-day money back guarantee on all our products.

MEMOTECH

MEMOTECH LIMITED, WITNEY, OXON. OX8 6BX
TELEPHONE (0993) 2977 TELEX 83372 MEMTEC G

Diary

Chris Wilder, a salesman from Australia, has written a program he finds particularly useful for noting past events and future commitments.



Diary is a user-friendly program written for the ZX81 with 16K RAM Pack.

When the program is RUN, line 80 displays a pleasantly-formatted menu providing a choice of six items. Item 1 produces a prompt for the entry of data couples which should be entered one couple at a time. The data couples should be entered in the form of a 10 character name and an eight digit code — this can be modified to take date and time. Entry must be in the form of name (A$) followed by Newline, and date (Z$) followed by Newline. A prompt in line 1150 has been included to avoid returning to the menu each time a data couple is entered. Provision is made in lines 10 and 20 for 200 data couples to be entered.

Item 2 gives a scrolled display of all the items currently entered plus a warning that the menu is about to be returned after completion of the scrolling. The menu is then returned.

Item 3 prints a list of the items *and* then returns to the menu. If you have much fewer than 200 items of data, it's a wise move to BREAK execution at this point and GOTO 80 as otherwise you'll use up a lot of paper for nothing!

Item 4 finds a name. A cue is presented asking you to input the name to be found. The search is made based on the initial three letters of the input name; for example, 'BILLY' will turn up the names 'BILL', 'BILKO', 'BILLS' and 'BILLY', so don't be surprised if you get more names than you expect. You will be informed of a failed search as well as a completed search, but with 200 items to search through this search may take some time. When you have found your chosen name, you have the option of copying the information onto a printer or else returning to the menu. If COPY is chosen, the computer will return to the menu once the printing has been accomplished. When a given name or names are found, the corresponding dates of entry are always displayed next to the names.

Item 5 does more or less the same as Item 4, but it will search for a given date rather than the name. Once found, the date will be displayed first, followed by the names entered on that date.

Item 6 saves the program to tape after setting up a safeguard in the form of a question which requires the answer yes (Y) or no (N) before it will execute a save. The computer then displays a 'save begins' message and starts to execute the save.

A note in your diary

The program must not be RUN when full or nearly full of data. Always GOTO 80. This should be overcome when saving from within the program since this automatically returns the menu when re-loaded.

As this program is a diary, you are able to make a few entries every day saving, adding and saving again until all 200 entries have been made or your chosen period has elapsed.

The program is aimed at those of you who have to make a record of past activities, although it could also be used to keep track of future commitments. Chris Wilder, the author of the program, used the program to keep a record of the customers he visits. The 'find name' option in Item 4 is very useful for Chris as he often wants to know when the last time was he saw a particular customer.

It is advisable to leave out redundant zeros when entering dates and to use full stops instead of dashes (simply because it's quicker to enter dots on the ZX81! The abbreviated dates will make data searches much quicker as well.

Extremely simple in concept, with the multitude of prompts and automatic returns, Diary should be able to operated by anyone with a minimum of instruction.

```
   5 REM "DIARY"
  10 DIM A$(200,10)
  20 DIM Z$(200,8)
  30 LET A$(200)="END"
  40 LET Z$(200)="00/00/00"
  50 LET B$=""
  70 CLS
  80 PRINT "**MENU**",,,,"ENTER
A NUMBER",,,,"1) MAKE ENTRY",,"2
) DISPLAY LIST",,"3) PRINT LIST"
,,"4) FIND NAME",,"5) FIND DATE"
,,"6) SAVE"
  90 INPUT A
 100 GOTO A*1000
1000 CLS
1010 LET X=1
1020 IF A$(X, TO 3)="END" THEN G
OTO 1070
1030 IF A$(X)=B$ THEN GOTO 1100
1040 LET X=X+1
1050 GOTO 1020
1070 PRINT "NO MORE ROOM"
1100 CLS
1110 PRINT "ENTER NAME AND DATE."
"
1120 INPUT A$(X)
1130 INPUT Z$(X)
1140 CLS
1150 PRINT "MORE DATA? (Y/N)"
1200 INPUT E$
1250 IF E$="Y" THEN GOTO 1000
1300 IF E$="N" THEN GOTO 80
2000 CLS
2010 LET X=0
2020 LET X=X+1
2030 SCROLL
2040 PRINT A$(X);"  ";Z$(X)
2050 IF A$(X, TO 3)<>"END" THEN
GOTO 2020
2060 PAUSE 300
2065 CLS
2070 PRINT "I AM NOW RETURNING Y
OU TO THE    MENU"
2080 PAUSE 200
2090 CLS
2100 GOTO 80
3000 CLS
3010 LPRINT """DIARY"""
3020 LPRINT
3030 LET X=0
3040 LET X=X+1
3050 LPRINT A$(X);"  ";Z$(X)
3060 IF A$(X)<>"END" THEN GOTO 3
040
3070 GOTO 80
4000 CLS
4010 PRINT "NAME, PLEASE"
4020 INPUT D$
4030 IF LEN D$<3 THEN GOTO 4020
4040 LET P=0
4050 LET X=1
4060 CLS
4070 IF A$(X, TO 3)<>D$( TO 3) T
HEN GOTO 4100
4080 LET P=1
4090 PRINT A$(X);"  ";Z$(X)
4100 LET X=X+1
4150 IF A$(X, TO 3)<>"END" THEN
GOTO 4070
4160 IF P=0 THEN PRINT D$;" NOT
FOUND"
4165 PRINT
4170 PRINT "MENU OR COPY?"
4180 INPUT M$
4190 IF M$="M" THEN GOTO 80
4200 IF M$="C" THEN COPY
4205 CLS
4210 GOTO 80
5000 CLS
5010 PRINT "DATE, PLEASE"
5020 INPUT U$
5030 IF LEN U$<3 THEN GOTO 5020
5040 LET P=0
5050 LET X=1
5060 CLS
5070 IF Z$(X, TO 3)<>U$( TO 3) T
HEN GOTO 5100
5080 LET P=1
5090 PRINT Z$(X);"  ";A$(X)
5100 LET X=X+1
5150 IF Z$(X, TO 3)<>"00/" THEN
GOTO 5070
5160 IF P=0 THEN PRINT U$;" NOT
FOUND"
5165 PRINT
5170 PRINT "MENU OR COPY?"
5180 INPUT C$
5190 IF C$="M" THEN GOTO 80
5200 IF C$="C" THEN COPY
5210 GOTO 80
6000 CLS
6010 PRINT "**ARE YOU READY TO E
XECUTE"
6020 PRINT "SAVE?**"
6030 INPUT U$
6040 IF U$="Y" THEN GOTO 6050
6050 PRINT AT 10,11;"SAVING BEGI
NS"
6055 PAUSE 80
6056 SAVE "DIARY"
6060 CLS
6070 GOTO 80
```

Sinclair ZX Spect

16K or 48K RAM... full-size moving-key keyboard... colour and sound... high-resolution graphics...

From only £125!



First, there was the world-beating Sinclair ZX80. The first personal computer for under £100.

Then, the ZX81. With up to 16K RAM available, and the ZX Printer. Giving more power and more flexibility. Together, they've sold over 500,000 so far, to make Sinclair world leaders in personal computing. And the ZX81 remains the ideal low-cost introduction to computing.

Now there's the ZX Spectrum! With up to 48K of RAM. A full-size moving-key keyboard. Vivid colour and sound. High-resolution graphics. And a low price that's unrivalled.

Professional power–personal computer price!

The ZX Spectrum incorporates all the proven features of the ZX81. But its new 16K BASIC ROM dramatically increases your computing power.

You have access to a range of 8 colours for foreground, background and border, together with a sound generator and high-resolution graphics.

You have the facility to support separate data files.

You have a choice of storage capacities (governed by the amount of RAM). 16K of RAM (which you can uprate later to 48K of RAM) or a massive 48K of RAM.

Yet the price of the Spectrum 16K is an amazing £125! Even the popular 48K version costs only £175!

You may decide to begin with the 16K version. If so, you can still return it later for an upgrade. The cost? Around £60.

Ready to use today, easy to expand tomorrow

Your ZX Spectrum comes with a mains adaptor and all the necessary leads to connect to most cassette recorders and TVs (colour or black and white).

Employing Sinclair BASIC (now used in over 500,000 computers worldwide) the ZX Spectrum comes complete with two manuals which together represent a detailed course in BASIC programming. Whether you're a beginner or a competent programmer, you'll find them both of immense help. Depending on your computer experience, you'll quickly be moving into the colourful world of ZX Spectrum professional-level computing.

There's no need to stop there. The ZX Printer–available now– is fully compatible with the ZX Spectrum. And later this year there will be Microdrives for massive amounts of extra on-line storage, plus an RS232/network interface board.



Key features of the Sinclair ZX Spectrum

- Full colour–8 colours each for foreground, background and border, plus flashing and brightness-intensity control.
- Sound–BEEP command with variable pitch and duration.
- Massive RAM–16K or 48K.
- Full-size moving-key keyboard–all keys at normal typewriter pitch, with repeat facility on each key.
- High-resolution–256 dots horizontally x 192 vertically, each individually addressable for true high-resolution graphics.
- ASCII character set–with upper- and lower-case characters.
- Teletext-compatible–user software can generate 40 characters per line or other settings.
- High speed LOAD & SAVE–16K in 100 seconds via cassette, with VERIFY & MERGE for programs and separate data files.
- Sinclair 16K extended BASIC–incorporating unique 'one-touch' keyword entry, syntax check, and report codes.



The ZX Printer – available now

Designed exclusively for use with the Sinclair ZX range of computers, the printer offers ZX Spectrum owners the full ASCII character set – including lower-case characters and high-resolution graphics.

A special feature is COPY which prints out exactly what is on the whole TV screen without the need for further instructions. Printing speed is 50 characters per second, with 32 characters per line and 9 lines per vertical inch.

The ZX Printer connects to the rear of your ZX Spectrum. A roll of paper (65ft long and 4in wide) is supplied, along with full instructions. Further supplies of paper are available in packs of five rolls.



The ZX Microdrive – coming soon

The new Microdrives, designed especially for the ZX Spectrum, are set to change the face of personal computing by providing mass on-line storage.

Each Microdrive can hold up to 100K bytes using a single interchangeable storage medium.

The transfer rate is 16K bytes per second, with an average access time of 3.5 seconds. And you'll be able to connect up to 8 Microdrives to your Spectrum via the ZX Expansion Module.

A remarkable breakthrough at a remarkable price. The Microdrives will be available in the early part of 1983 for around £50.



ZX Spectrum software on cassettes – available now

The Spectrum software library is growing every day. Subjects include games, education, and business/household management. Flight Simulation...Chess...Planetoids...History...Inventions...VU-CALC...VU-3D ...Club Record Controller...there is something for everyone. And they all make full use of the Spectrum's colour, sound, and graphics capabilities. You'll receive a detailed catalogue with your Spectrum.

ZX Expansion Module

This module incorporates the three functions of Microdrive controller, local area network, and RS232 interface. Connect it to your Spectrum and you can control up to eight Microdrives, communicate with other computers, and drive a wide range of printers.

The potential is enormous, and the module will be available in the early part of 1983 for around £30.

sinclair

Sinclair Research Ltd, Stanhope Road, Camberley, Surrey GU15 3PS. Tel: Camberley (0276) 685311.

How to order your ZX Spectrum

BY PHONE – Access, Barclaycard or Trustcard holders can call 01-200 0200 for personal attention 24 hours a day, every day. BY FREEPOST – use the no-stamp needed coupon below. You can pay by cheque, postal order, Barclaycard, Access or Trustcard.

EITHER WAY – please allow up to 28 days for delivery. And there's a 14-day money-back option, of course. We want you to be satisfied beyond doubt – and we have no doubt that you will be.

To: Sinclair Research, FREEPOST, Camberley, Surrey, GU15 3BR. **Order**

Qty	Item	Code	Item Price £	Total £
	Sinclair ZX Spectrum – 16K RAM version	100	125.00	
	Sinclair ZX Spectrum – 48K RAM version	101	175.00	
	Sinclair ZX Printer	27	59.95	
	Printer paper (pack of 5 rolls)	16	11.95	
	Postage and packing: orders under £100	28	2.95	
	orders over £100	29	4.95	
			Total £	

Please tick if you require a VAT receipt ☐

*I enclose a cheque/postal order payable to Sinclair Research Ltd for £ ______

*Please charge to my Access/Barclaycard/Trustcard account no.

*Please delete/complete as applicable

Signature

PLEASE PRINT

Name: Mr/Mrs/Miss

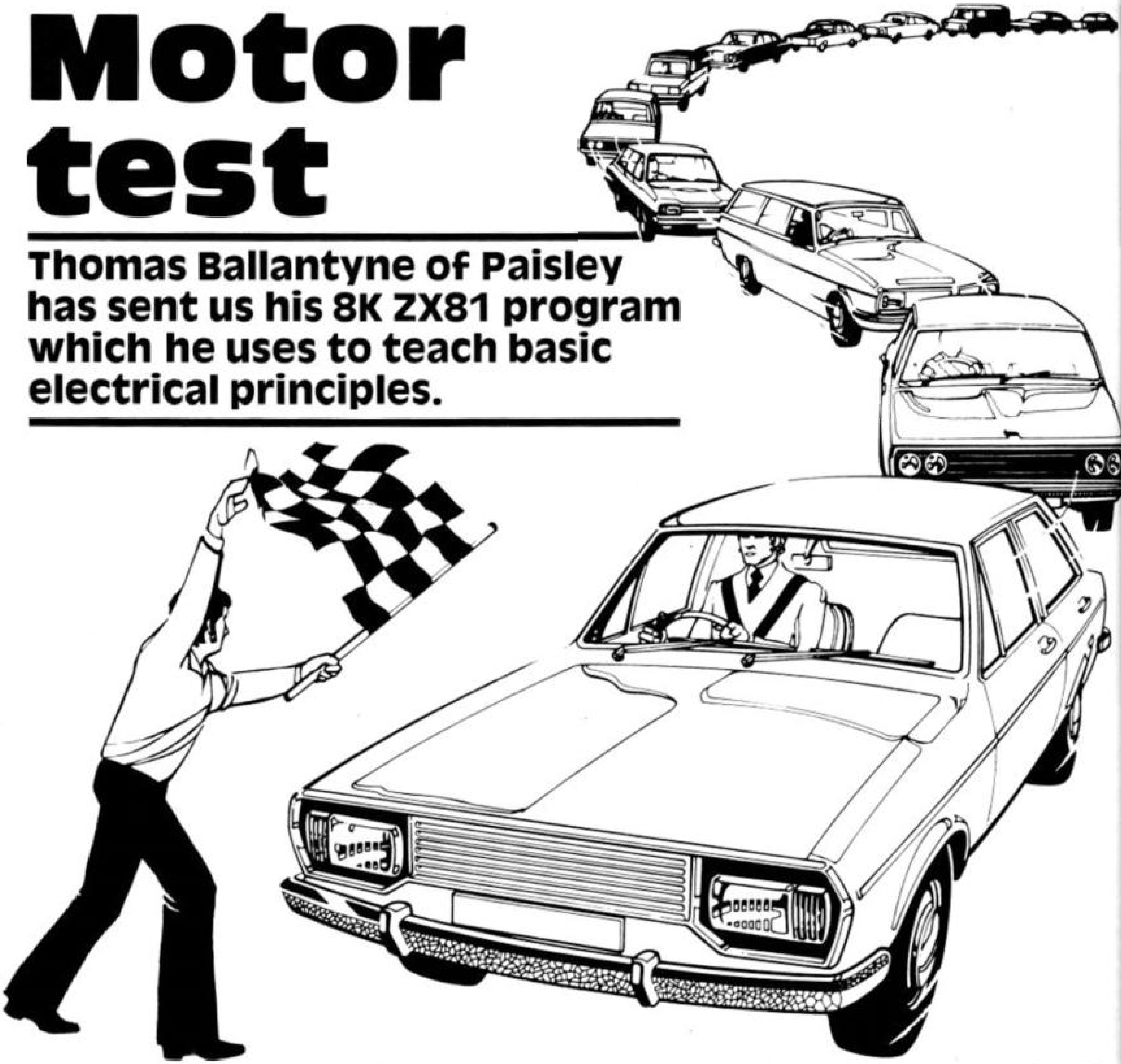
Address

ZXC 902

FREEPOST – no stamp needed. Prices apply to UK only. Export prices on application.

Motor test

Thomas Ballantyne of Paisley has sent us his 8K ZX81 program which he uses to teach basic electrical principles.



This program was devised to simulate the electrical conditions which occur when a single phase AC motor is in operation.

The first section of the program draws a circuit diagram which represents the motor connected to an electrical supply, with instruments in circuit to measure current, voltage and the input power. Readings of these values and also of the power output of the motor are shown in the diagram. These values appear when the motor is operated.

Since the program is used in the teaching of basic electrical principles, it now asks you to enter the values of four quantities. These are:

(a) The motor efficiency.
(b) The motor power factor.
(c) The input Volt Amperes.
(d) The reactive power.

Information can be obtained from the program in order to devise a solution to these problems. If you want to delve a bit more deeply into the electrical theory involved, then an electrical text book must be consulted.

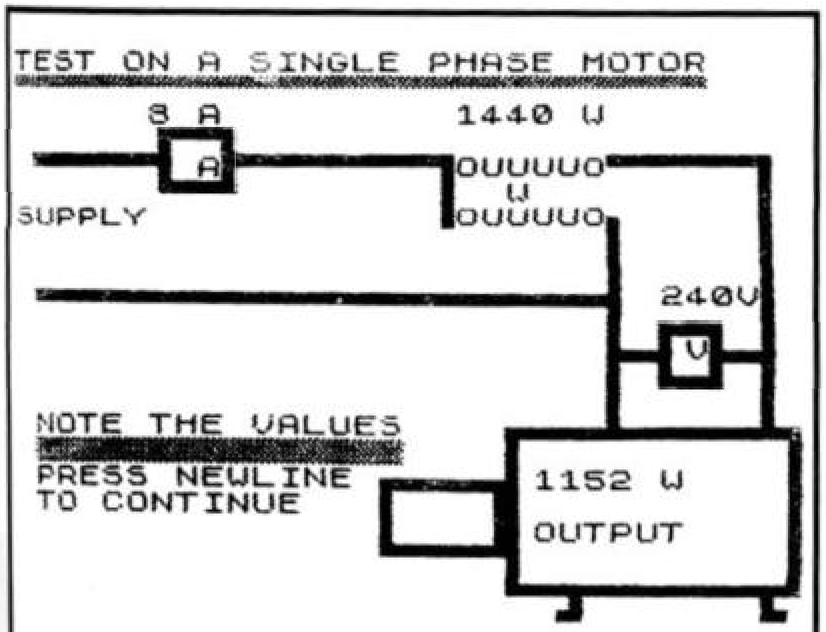
A teaching program should be designed to correct wrong answers. Entry of a wrong answer causes the program to display how the correct one should be obtained. A further entry of a wrong answer causes the correct solution, and the method of achieving it to appear.

The motor takes a random current of between 1 and 10 Amperes. The supply voltage is constant at 240V. This voltage is chosen because it is the standard value of the single phase supply in this country. The current is limited to simulate practical conditions; if the voltage was to be higher, then a three phase system of supply would have to be used.

Motor ways

Lines 10 to 980 print the diagram and set the variables. If a different voltage, say 110V, is required, simply alter line 750. When the program is run, lines 1000 to 1130 generate the questions to be answered, and the methods to be used. The remainder of the program is devoted to displaying the answers entered and to showing the user how to correct them, if necessary.

There is no way round the fact that a program of this type is rather long (this one takes about 8K) as every eventuality must be catered for. The disadvantage of any teaching program is the amount of memory that is used. Once it is made up, however, it can be used over and over again. It often also forms the basis for a new program at a later time.



A sample screen display showing a circuit of a single phase motor.

```
   1 REM "MOTOR TEST"
   2 REM T.BALLANTYNE
   9 SLOW
  10 PRINT "TEST ON A SINGLE PHA
SE MOTOR"
  20 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒"
  30 FOR X=2 TO 12
  40 PLOT X,35
  50 NEXT X
  60 FOR Y=33 TO 37
  70 PLOT 13,Y
  75 PLOT 18,Y
  80 NEXT Y
  90 FOR X=14 TO 17
 100 PLOT X,33
 110 PLOT X,37
 120 NEXT X
 130 FOR X=18 TO 36
 140 PLOT X,35
 150 NEXT X
 160 PRINT AT 4,16;"OUUUUO"
 170 FOR X=48 TO 60
 180 PLOT X,35
 190 NEXT X
 200 PRINT AT 6,16;"OUUUUO"
 210 FOR Y=30 TO 35
 220 PLOT 35,Y
 230 NEXT Y
 240 FOR X=2 TO 48
 250 PLOT X,24
 260 NEXT X
 270 FOR Y=14 TO 30
 280 PLOT 48,Y
 282 NEXT Y
 285 FOR Y=14 TO 35
 290 PLOT 60,Y
 300 NEXT Y
 310 FOR X=52 TO 56
 320 PLOT X,22
 330 PLOT X,18
 340 NEXT X
 350 FOR Y=18 TO 22
 360 PLOT 52,Y
 370 PLOT 56,Y
 380 NEXT Y
 390 FOR X=48 TO 52
 400 PLOT X,20
 420 NEXT X
 430 FOR X=56 TO 60
 440 PLOT X,20
 450 NEXT X
 460 PRINT AT 6,0;"SUPPLY"
 462 PRINT AT 4,8;"A"
 464 PRINT AT 11,27;"V"
 466 PRINT AT 5,20;"W"
 470 FOR X=40 TO 62
 480 PLOT X,14
 490 PLOT X,2
 500 NEXT X
 510 FOR Y=2 TO 14
 520 PLOT 62,Y
 530 NEXT Y
 540 PRINT AT 21,30;"▙"
 550 PRINT AT 21,22;"▟"
 560 FOR Y=2 TO 14
 570 PLOT 40,Y
 580 NEXT Y
 590 FOR Y=5 TO 9
 600 PLOT 30,Y
 610 NEXT Y
 620 FOR X=30 TO 40
 630 PLOT X,10
 640 PLOT X,5
 650 NEXT X
 660 FOR Y=6 TO 10
 670 PLOT 39,Y
 680 NEXT Y
 682 PRINT AT 14,1;"PRESS NEWLIN
E "
 683 PRINT AT 15,2;"TO START MOT
OR"
 684 INPUT D$
 685 PRINT AT 14,1;"
"
 686 PRINT AT 15,2;"
"
 687 GOSUB 740
 688 FOR Z=1 TO 100
 690 PRINT AT 18,16;"▒▒▒▒"
 700 PRINT AT 18,16;"    "
 710 PRINT AT 17,16;"▒▒▒▒"
 720 PRINT AT 17,16;"    "
 730 NEXT Z
 735 GOTO 950
 740 LET A=INT (RND*10)+1
 750 LET V=240
 760 IF A=1 THEN LET PF=0.4
 765 IF A=1 THEN LET EF=0.4
 770 IF A=2 THEN LET PF=0.45
 775 IF A=2 THEN LET EF=0.5
 780 IF A=3 THEN LET PF=0.5
 785 IF A=3 THEN LET EF=0.55
 790 IF A=4 THEN LET PF=0.55
 795 IF A=4 THEN LET EF=0.6
 800 IF A=5 THEN LET PF=0.6
 802 IF A=5 THEN LET EF=0.65
 805 IF A=6 THEN LET PF=0.65
 810 IF A=6 THEN LET EF=0.7
 815 IF A=7 THEN LET PF=0.7
 820 IF A=7 THEN LET EF=0.75
 830 IF A=8 THEN LET PF=0.75
 835 IF A=8 THEN LET EF=0.8
 840 IF A=9 THEN LET PF=0.8
 845 IF A=9 THEN LET EF=0.85
 850 IF A=10 THEN LET PF=0.9
 855 IF A=10 THEN LET EF=0.8
 860 LET W=V*A*PF
 870 LET O=V*A*PF*EF
 880 LET S=V*A
 890 LET X=ACS (PF)
 900 LET Q=V*A*SIN X
 910 PRINT AT 2,18;W;" W"
 920 PRINT AT 16,21;O;" W"
 925 PRINT AT 18,21;"OUTPUT"
 930 PRINT AT 9,26;V;"V"
 940 PRINT AT 2,6;A;" A"
 945 RETURN
 950 PRINT AT 14,1;"NOTE THE VAL
UES"
 960 PRINT AT 15,1;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒"
 970 PRINT AT 16,1;"PRESS NEWLIN
E"
 980 PRINT AT 17,1;"TO CONTINUE"
 990 INPUT X$
1000 CLS
1010 PRINT "CALCULATE THE FOLLOW
ING"
1015 PRINT "ENTER YOUR ANSWERS"
1016 PRINT
```

```
1020 PRINT "EFFICIENCY OF MOTOR=?"
1021 PRINT
1022 GOSUB 4000
1023 INPUT A$
1024 IF A$="YES" THEN GOTO 2000
1025 GOSUB 4020
1026 INPUT E1
1027 CLS
1028 GOSUB 4040
1030 PRINT
1040 PRINT "POWER FACTOR OF MOTOR=?"
1041 PRINT
1042 GOSUB 4000
1043 INPUT B$
1044 IF B$="YES" THEN GOTO 2500
1046 CLS
1047 GOSUB 4020
1048 INPUT P1
1049 CLS
1050 GOSUB 4040
1051 PRINT
1052 GOSUB 4080
1060 PRINT
1070 PRINT "VA INPUT=?"
1071 PRINT
1072 GOSUB 4000
1073 INPUT C$
1074 IF C$="YES" THEN GOTO 3000
1075 CLS
1077 GOSUB 4020
1078 INPUT S1
1079 CLS
1080 GOSUB 4040
1085 PRINT
1090 GOSUB 4080
1092 PRINT
1094 GOSUB 4100
1096 PRINT
1100 PRINT "REACTIVE POWER IN VAR=?"
1110 PRINT "TO NEAREST WHOLE NUMBER"
1115 PRINT
1116 GOSUB 4000
1117 INPUT D$
1118 IF D$="YES" THEN GOTO 3500
1119 CLS
1120 GOSUB 4020
1121 INPUT Q1
1122 CLS
1123 GOSUB 4040
1124 PRINT
1125 GOSUB 4080
1126 PRINT
1127 GOSUB 4100
1128 PRINT
1129 GOSUB 4120
1130 PRINT
1132 PRINT " PRESS NEWLINE TO CONTINUE "
1134 INPUT Z$
1135 IF NOT E1=EF THEN GOSUB 1400
1140 IF NOT P1=PF THEN GOSUB 1500
1150 IF NOT S1=S THEN GOSUB 1600
1160 IF NOT INT (Q1+0.5)=INT (Q+0.5) THEN GOSUB 1700
1165 CLS
1170 PRINT "WELL DONE"
1175 PRINT "         "
1176 PAUSE 150
1177 POKE 16437,255
1178 CLS
1179 PRINT
1180 PRINT "CORRECT ANSWERS ARE"
1185 PRINT
1190 PRINT "EFFICIENCY=";EF
1195 PRINT
1200 PRINT "POWER FACTOR=";PF;"LAG"
1205 PRINT
1210 PRINT "VA INPUT=";S;"VA"
1215 PRINT
1220 PRINT "REACTIVE POWER=";INT (Q+0.5);"VAR"
1225 PRINT
1230 PRINT "PROBLEM COMPLETED"
1235 PRINT
1240 PRINT "DO YOU WANT TO TRY AGAIN YES/NO"
1250 INPUT Z$
1260 IF Z$="YES" THEN GOTO 1300
1265 CLS
1270 PRINT
1280 PRINT "OK TRY AGAIN SOME TIME"
```




```
1290 STOP
1300 CLS
1310 GOTO 10
1400 CLS
1401 PRINT "EFFICIENCY"
1402 PRINT
1404 PRINT "YOUR ANSWER ";E1;" IS NOT CORRECT"
1405 PRINT
1406 PRINT "USE THIS METHOD"
1407 PRINT
1408 PRINT "EFFIC.=OUTPUT/INPUT"
1409 PRINT
1410 PRINT "ALTER YOUR ANSWER"
1415 PRINT
1420 INPUT E1
1425 IF E1=EF THEN GOSUB 1804
1430 IF NOT E1=EF THEN GOSUB 1800
1440 LET E1=EF
1450 RETURN
1500 CLS
1501 PRINT "POWER FACTOR"
1502 PRINT
1504 PRINT "YOUR ANSWER ";P1;" IS NOT CORRECT"
1505 PRINT
1506 PRINT "USE THIS METHOD"
1507 PRINT
1508 PRINT "POWER FACTOR=INPUT POWER/VA"
1509 PRINT
1510 PRINT "ALTER YOUR ANSWER"
1515 PRINT
1520 INPUT P1
1525 IF P1=PF THEN GOSUB 1852
1530 IF NOT P1=PF THEN GOSUB 1850
1540 LET P1=PF
1550 RETURN
1600 CLS
1601 PRINT "VA INPUT"
1602 PRINT
1606 PRINT "YOUR ANSWER ";S1;" IS INCORRECT"
1607 PRINT
1608 PRINT "VA INPUT=VOLTS*AMPS"
1609 PRINT
1610 PRINT "ALTER YOUR ANSWER"
1615 PRINT
1620 INPUT S1
```

```
1625 IF S1=S THEN GOSUB 1903
1630 IF NOT S1=S THEN GOSUB 1900
1640 LET S1=S
1650 RETURN
1700 CLS
1704 PRINT "REACTIVE POWER"
1705 PRINT
1706 PRINT "YOUR ANSWER ";Q1;" IS INCORRECT"
1707 PRINT
1708 PRINT "REACTIVE POWER=VA SIN G"
1709 PRINT
1710 PRINT "ALTER YOUR ANSWER"
1715 PRINT
1720 INPUT Q1
1725 IF Q1=Q THEN GOSUB 1958
1730 IF NOT Q1=Q THEN GOSUB 1950
1740 LET Q1=Q
1750 RETURN
1800 PRINT
1801 PRINT "YOUR ANSWER IS STILL INCORRECT"
1802 PRINT
1804 PRINT "ANSWER TO EFFICIENCY ="
1805 PRINT "OUTPUT/INPUT=";O;"/";W;"=";EF
1806 PRINT
1807 PRINT
1808 PRINT "PRESS NEWLINE TO CONTINUE"
1809 INPUT Z$
1810 RETURN
1850 PRINT
1851 PRINT "YOUR ANSWER IS STILL INCORRECT"
1852 PRINT
1853 PRINT "ANSWER TO POWER FACTOR="
1854 PRINT "INPUT WATTS/VOLTS XAMPS="
1855 PRINT W;"/";V;"*";A;"=";PF;" LAG"
1856 PRINT
1857 PRINT
1858 PRINT "PRESS NEWLINE TO CONTINUE"
1859 INPUT Y$
1860 RETURN
1900 PRINT
1902 PRINT "YOUR ANSWER IS STILL INCORRECT"
1904 PRINT "ANSWER TO VA INPUT=";V;"*";A;"=";S;" VA."
1905 PRINT
1906 PRINT
1907 PRINT "PRESS NEWLINE TO CONTINUE"
1908 INPUT X$
1910 RETURN
1950 PRINT
1957 PRINT "YOUR ANSWER IS STILL INCORRECT"
1958 PRINT
1959 PRINT "ANSWER TO REACTIVE "
1960 PRINT "POWER= V * A * SIN G"
1961 PRINT
1962 PRINT "WHERE G IS THE PHASE ANGLE"
1964 PRINT
1966 PRINT "G=ACS PF=";X*180/PI;" DEGS."
1968 PRINT
1970 PRINT "REACTIVE POWER=";V;"*";A;"*";SIN (X);"=";INT (Q+0.5);" VAR."
1974 PRINT "PRESS NEWLINE TO CONTINUE"
1976 INPUT U$
1980 RETURN
2000 PRINT
2010 PRINT "EFFIC.=OUTPUT WATTS"
2020 PRINT "                ▀▀▀▀▀▀▀▀▀▀▀"
2030 PRINT "          INPUT WATTS"
2040 PRINT
2050 PRINT "NOW WHATS THE ANSWER ?"
2060 GOTO 1026
2500 PRINT
2510 PRINT "COS G=P.F.=INPUT WATTS"
2520 PRINT "                ▀▀▀▀▀▀▀▀▀▀▀"
2530 PRINT "            VOLTS X AMPS."
2540 PRINT
2550 PRINT "NOW WHATS THE ANSWER ?"
2560 GOTO 1048
3000 PRINT
3010 PRINT "VA INPUT=VOLTS X AMPS"
3020 PRINT
3030 PRINT "NOW WHATS THE ANSWER ?"
3040 GOTO 1078
3500 PRINT
3510 PRINT "REACTIVE POWER=VA SIN G"
3520 PRINT "WHERE G IS THE PHASE ANGLE"
3530 PRINT
3540 PRINT "NOW WHATS THE ANSWER ?"
3550 GOTO 1121
4000 PRINT "WANT MORE INFORMATION ? YES/NO"
4010 RETURN
4020 PRINT
4025 PRINT " ENTER YOUR ANSWER "
4030 RETURN
4040 PRINT "        YOUR ANSWERS "
4050 PRINT
4060 PRINT "EFFIC.OF MOTOR=";E1
4070 RETURN
4080 PRINT "POWER FACTOR OF MOTOR=";P1;" LAG"
4090 RETURN
4100 PRINT "VA INPUT=";S1;" VA."
4110 RETURN
4120 PRINT "REACTIVE POWER=";Q1;" VAR."
4130 RETURN
```

```
CALCULATE THE FOLLOWING
ENTER YOUR ANSWERS

EFFICIENCY OF MOTOR=?

WANT MORE INFORMATION ? YES/NO

EFFIC.=OUTPUT WATTS
       ▀▀▀▀▀▀▀▀▀▀▀▀
        INPUT WATTS

NOW WHATS THE ANSWER ?


          YOUR ANSWERS

EFFIC.OF MOTOR=0.8

POWER FACTOR OF MOTOR=0.8 LAG

VA INPUT=1200 VA.

REACTIVE POWER=900 VAR.

PRESS NEWLINE TO CONTINUE
```

If you can't solve the problem, the program will first offer you some help and then, if you still can't manage it, show you how to do it.

In search of adventure



Now that the Christmas rush is over, Paul Holmes takes a look at some of the latest software for the ZX Spectrum.

I expect that by now you are as sick of turkey sandwiches and Christmas cake as I am, and are now on the lookout for some post-festive inspiration for your ZX Spectrum. Whether you managed to survive the Christmas period with some cash or you are wondering what to do with that fiver you received from your Auntie, here are a number of goodies to choose from.

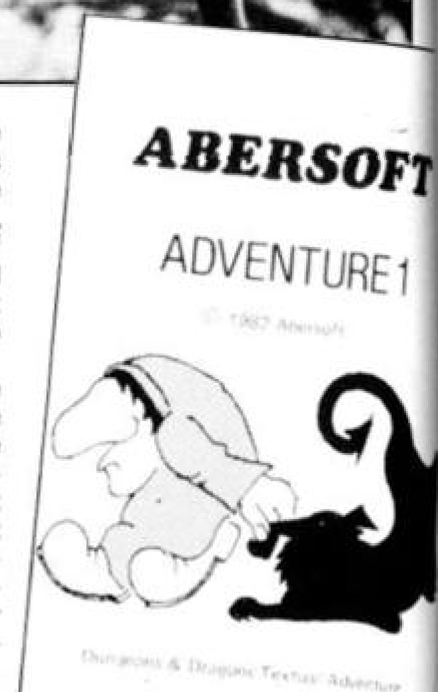
Adventure 1 — Abersoft

This is an adventure game, and for the uninitiated an adventure game is one of a 'search' usually for treasure of some description hidden in caves, passages, jungles, Egyptian pyramids and the like. Just to make matters more complicated, there are usually a number of monsters and mysterious objects whose sole aim in life is to hinder or help you on your quest.

Abersoft's adventure, however, has no harmful monsters but makes you rely on your own intelligence to get around the passages and utilise the objects you find to help you win the treasure. In the first scene, you are looking at a water house and you can see a large set of keys, a shiny brass lamp, some tasty food and a bottle of water. You can instruct the computer by giving it simple one or two word instructions such as 'get keys' or 'take lamp'. Using these keywords, you can pick up a number of items that you might think will be of use to you at a later point in the game; you are allowed to 'hold' up to eight items.

To find out what you are holding at any stage in the game, you type 'inventory'; the computer's recognition of vocabulary is very good, and fast too due to the fact that the vast majority of the program is written in machine code. The computer only scans the first four letters of each keyword so 'inve' is the same as 'inventory'.

To move about in the game,



simple compass directions should be given such as 'N', 'E', 'S' and 'W'. Instructions such as 'enter' are obeyed such that if you are next to a building or the entrance to a passage, you will appear inside.

A 'smashing' game

The adventure is based on a series of caves, canyon crawls and passages. There are countless rooms with a white mist lingering on the floor (similar to a Top of the Pops rehearsal, maybe?). Most passages, rooms, etc, are empty but some will contain useful objects and treasure. The treasure varies from extremely heavy golden nuggets to easily breakable Ming vases.

The Ming vase had me in some confusion for a few days (don't get the idea that you'll manage to finish this game in a week!). The problem with the vase is that the only way to put something down is to use the command 'drop' which is countered with the reply 'you hurled it delicately to the ground'. (You have to put some of the objects down again as for every item of treasure you return to the water house, you receive 10 points.) So all I got the first few times I tried this was a smashed vase! Until I found the pillow, of course...

There are 21 items of treasure to find in the game in all, so don't expect an easy time. The adventure holds many secrets for the player to work out, one of which is the meaning of the secret messages on the walls in certain rooms. I managed to work out most of them, but I won't spoil your fun by telling you.

Lack of talent?

At one point in the game I came across a very strange room in which I was standing at a window looking down on a pit (complete with mist). Opposite me I could see a man at another window waving back at me. It has been a puzzle to me ever since — what is the relevance of the eerie figure? Perhaps he was, like me, another player driven to madness by this game.

I fear he might have been, as the game is non-graphical and requires great enthusiasm to play for more than a few hours at a stretch. Pure text, with no colour or sound, is a clear waste of the ZX Spectrum's talent and would clearly wear the interest of the player. (There is also a ZX81 version of the game available for £8.95.)

Far too often in the game, I found myself going round and round in circles. However, there's no resting to cook the odd meal or read a book or two while you await some happening — the computer always has a quick answer for you. For instance, in the depth of frustration, I told the computer 'where to go' to which it promptly told me 'Watch it. The wizard is watching you'. I found its understanding of my use of 'non-Queen's English' quite amusing and prompts me to applaud the author's sense of humour.

One of my main criticisms of the game is the actual quality of the cassettes themselves. One of the two copies I had for review had a header that was full of rubbish, so that the normally tolerant Spectrum rejected it. The whole program was littered with clicks and buzzes. However, I did manage to load the second copy, albeit with some difficulty.

Adventure is priced at £9.95 and is available from Abersoft, 7 Maes Afallen, Bow Street, Dyfed SY24 5BA.

Superview/Superdraw — Video Software

Video Software are now selling

```
You are standing at the end of a
road before a small brick
building. Around you is a
forest. A small stream flows
out of the building and down a
gully.

enter

You are inside a building, a
well house for a large spring.

About you can be seen
A large set of keys

A shiny brass lamp

Some tasty food

A bottle of water
```

```
You are at one end of a vast
hall stretching forward out of
sight to the west . There are
openings to either side .
Nearby, a wide stone staircase
leads downward. The hall is
filled with wisps of white mist
swaying to and fro almost as if
alive. A cold wind blows up the
staircase. There is a passage
at the top of a dome behind you.

About you can be seen

Rough stone steps leading up

gargle

I don't understand.
```

```
You can't get by the snake.

You are in the hall of the
mountain king, with passages off
in all directions.

About you can be seen

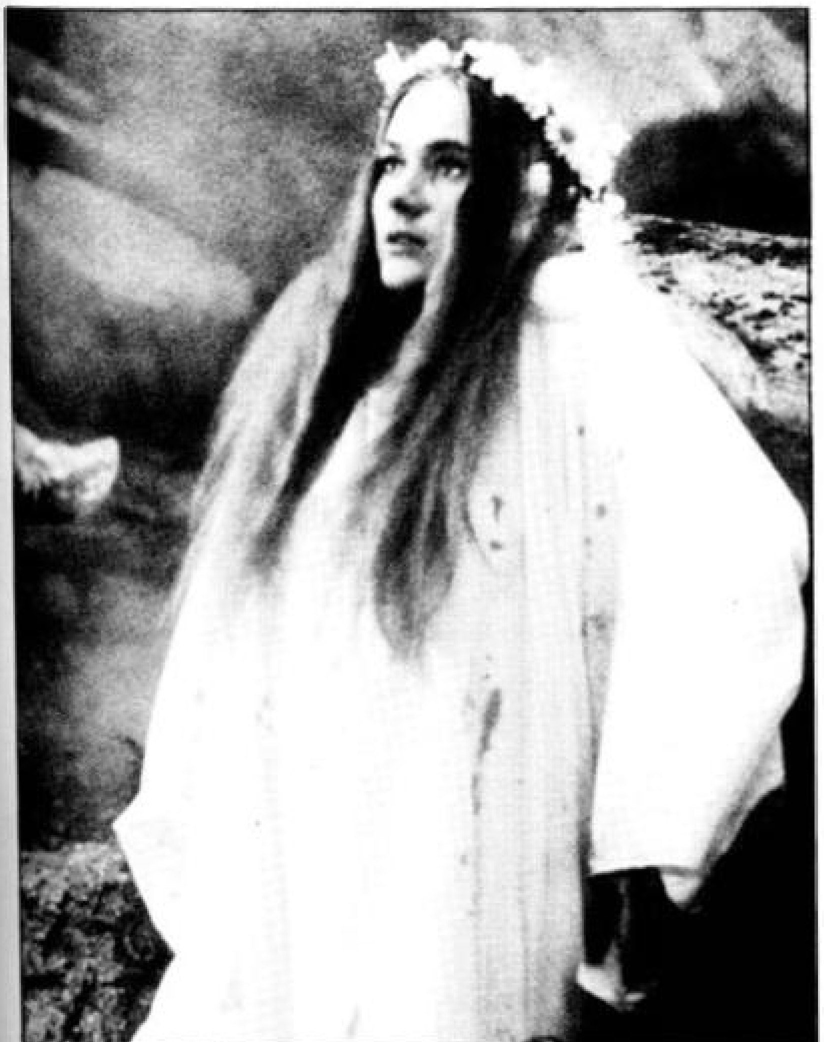
An aroused,huge,green snake,
barring the way

sing

No.I have a terrible voice

kill

Don't be ridiculous
```



two 'creative' packages for the 48K ZX Spectrum. The first is a text display and creating system called Superview.

Superview has two options: 42 pages of two colour text or 21 pages of eight colour text. These two options provide sensibly an optional trade-off between colour and space. The only thing that disappointed me was the waste of space in the colour storage methods. The most space saving system would have been to insert colour codes into the text wherever there was a change of INK, PAPER intensity and its FLASH mode. Instead, they have taken the option to store an individual attribute for each character, which is unnecessary. If they had taken the more efficient option, the storage size could have been doubled.

Superview is supplied on cassette, complete with a reasonable little manual held together by ring binders. Complementing the manual is a new concept of having a written commentary on the reverse side of the cassette. This is quite a good idea and should clearly be an indication to other manufacturers of software packages requiring good documentation.

What's on the menu?

On the tape, a demo set of 21 full colour pages are provided which can be loaded using Superview's 'Load a set of pages by name' option. These pages depict the history of computers and effectively show off the capabilities of good coloured text.

Superview operates via a menu which allows you to save and load pages, view pages in sequence, view pages on demand (in a similar way to Teletext and Prestel) and to create a new set of pages or alter the existing ones. To create a new set of pages, one must first destroy the existing ones leaving room for the new set. The pages are stored quite simply in character arrays and recalled to the screen using a very clever method. First of all, the screen attributes are set set to white so that the machine prints white characters onto a white background. Then, once a whole 'invisible' page has been printed, the attributes are transferred by machine code from an array to the attribute file. In this way the text appears to instantly arrive on the screen.

This program would be very useful to the shop owner who could set up some product information for his or her customers to inspect at their leisure. Overall, I was impressed.

Luck of the draw?

Superdraw is similar to Superview in that it is for visual creation, but Superdraw is for creating Hi-Res graphics. It too is based around a main menu and uses a similar programming style to its sister package.

In the drawing mode, full eight way cursor control is allowed along with a number of necessary functions to alter the colour, brightness, etc. Again, like Superview, there is a commentary on the reverse side of the cassette and the package is supplied with a ring bound manual which provides a full explanation of the system.

The loading problems with this tape were really very serious. I tried to get the program loaded using a 'new-fangled' graphic equaliser, but even then failed to get it into the machine. Because of this I am certainly not in a position to say whether or not you should buy this program. The best thing to do if you are interested in this particular tape is to arrange for some form of demonstration so that you can actually see it loaded into the machine.

Superview and Superdraw are priced £5.00 and are available from Video Software, Stone Lane, Kinver, Stourbridge, West Midlands DY7 6EQ.

```
      SUPERVIEW  MENU OF OPTIONS

1.   Clear or copy a page.

2.   Load a single page by name
3.   Load a set of pages by name.

4.   Save a single page by name.
5.   Save a set of pages by name.

6.   View pages on demand.
7.   View pages in sequence.

8.   Create/amend a page.
9.   Create a new set of pages
```

Purchase/sales ledger

Neil Streeter of Hastings decided to save himself some money – instead of buying a commercial purchase/sales ledger package, he thought he'd write one of his own!

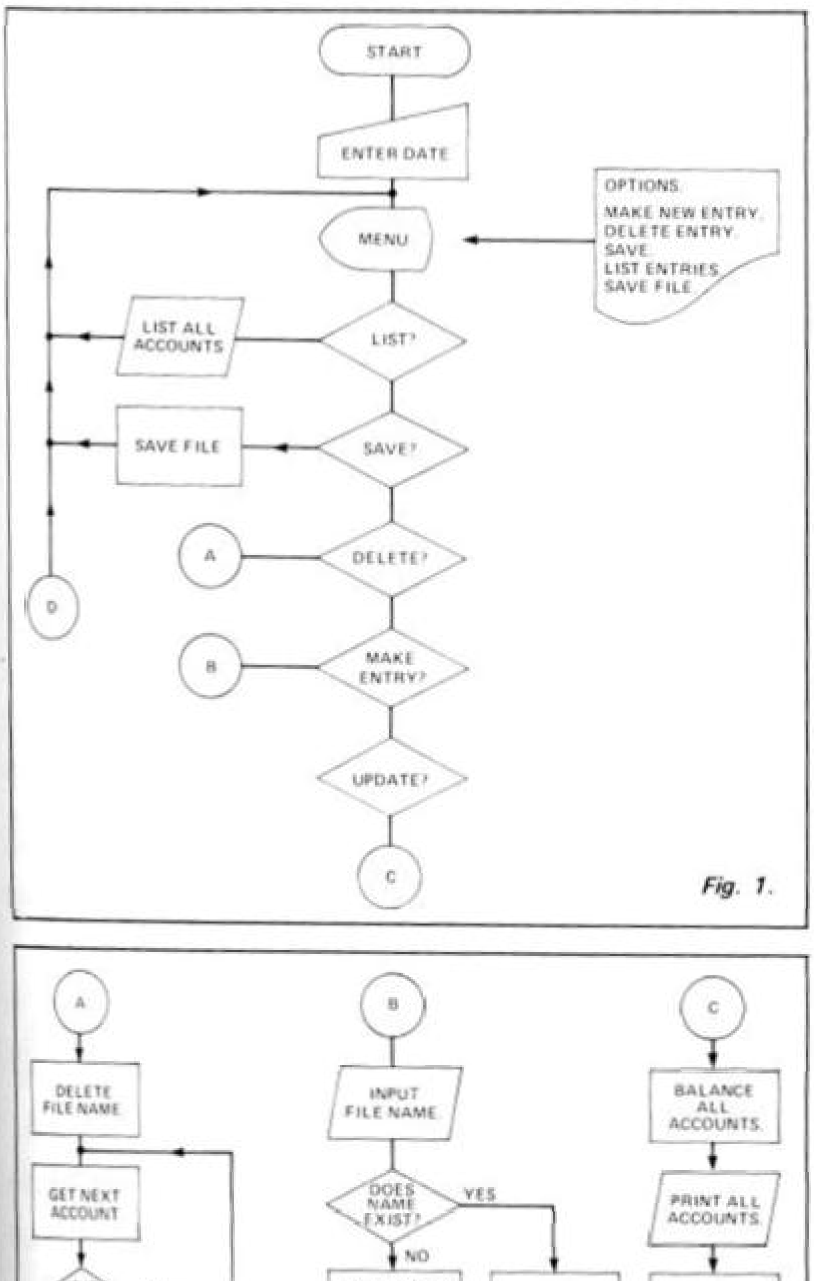


Fig. 1.

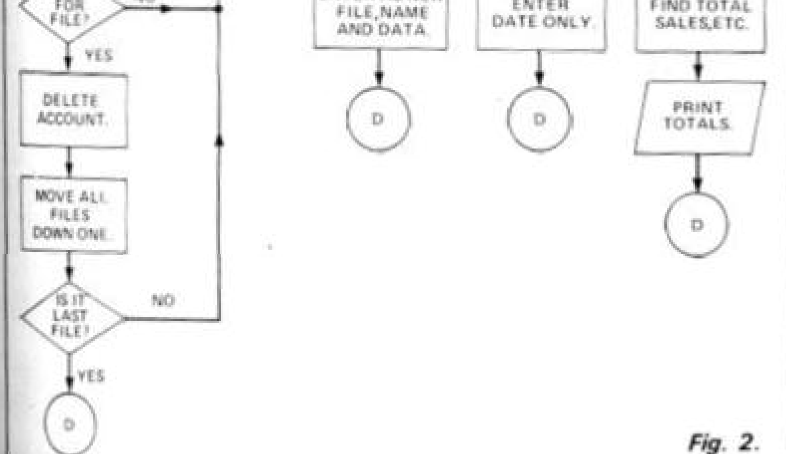


Fig. 2.



I recently became interested in getting hold of an accountancy package for my ZX81, especially those including purchase and sales ledger programs. However, the cost of these was slightly too high for my expenses (I am a bit of a miser I suppose). So, I decided to have a go at writing them myself. As the prices of these programs are in the range of £8-17, I had every reason to believe that this would be a difficult task as usually the more expensive programs reflect the complexity of the programming involved.

However, I did have one thing in my favour – I had studied basic accountancy while at the University of Aston in Birmingham, and I still had all my old text books. So I rescued them from their hiding place and, having dusted them off, looked through the indexes for a purchase and sales ledger.

I was amazed to find that it would not be as difficult as I had first imagined. One program would serve as the central core for both, with only minor alterations needed between the two. Basically (no pun intended), they consist of a number of files, each containing information on a sale or a purchase. At certain times, these accounts would have to be balanced and updated.

First things first though – I decided to draw up a series of flow charts to simulate exactly what the final program should do. These are shown in Figs. 1 and 2.

Setting up

Then, I set about the programming. My first decision involved the kind of format I was going to store the data in. I toyed with several ideas such as packed free, packed fixed, etc, until I decided to use a combination of the first two methods I found.

There would be two string arrays: the first would contain all the account names and the second, all the data. The first array was easy and just needed dimensioning. The second array required a packed fixed format – packed because the data was packed tightly and fixed because each piece of data always occupies the same number of bytes. The format for each string in the array is shown in Fig. 3.

AN was chosen to represent the account number. This would be a character, the code of which is the reference to the array of account names. VAT would be either the character I or E, depending on whether it was to represent 'inclusive' or 'exclusive'.

Thus, having worked out my format, flow diagrams and reference material, I set to the task of writing a program to fulfil my needs. It was finished after approximately seven hours work.

The program is quite straightforward to use and has space for 50 accounts and up to 300 entries. If you have less accounts you may alter the arrays to suit your requirements. The program listing as it stands is for the purchase ledger, but only six lines need be altered to magically transform it into a sales ledger. Two programs for the price of one – pretty good, eh?

I'm not saying that this program is as good as a commercial ▶

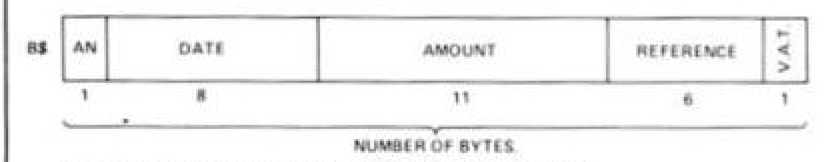


Fig. 3. The format for each string in the array.

package, but it does do the job and it's fairly fast too. Which may present a moral – just because the price is high, it doesn't mean you can't write it yourself.

Operation

The program, like all good business programs, is menu driven and extremely user friendly. It should be evident how to use the program as soon as you have it up and running.

However, when I set it up for the first time, I do the following. RUN the program, which should give you a date request on the screen. Then, EDIT and STOP, followed by Newline to break into the program. Now, enter:

LET DS(I) = "00:00:00"

as a direct command. Follow this with GOTO 1000 as a direct command.

You will now be requested to enter the date again which you should do and you will get the menu. You may now select the SAVE option and save the program either as a sales ledger or as a purchase ledger. This means that your program will automatically start on loading with the first date as 00:00:00.

References

Business Accounting 1 and 2 – Frank Wood, published by Longman.
Framework of Accountancy – CC Magee, published by MacDonald and Evans.

```
  10 DIM A$(50,20)
  20 DIM B$(300,27)
  30 DIM B(50,3)
  40 DIM D$(2,8)
  50 LET NA=0
  60 LET AC=0
1000 CLS
1010 PRINT "ENTER DATE AS:- (DD/
MM/YY)"
1020 INPUT E$
1030 IF LEN E$<>8 THEN GOTO 1020
1040 LET D$(2)=E$
1050 CLS
1060 PRINT D$(2)
1070 PRINT ,,"PURCHASE LEDGER OP
TIONS:-","
   ",,,"1. MAKE ENTRY."
1080 PRINT "2. LIST ACCOUNTS.","
3. DELETE AN ACCOUNT.","4. PRINT
 AN ACOUNT RECORD.","5. BALLANCE
 ACCOUNTS.","6. SAVE FILE."
1100 LET E$=INKEY$
1110 IF E$<"1" OR E$>"6" THEN GO
TO 1100
1130 GOTO 1000+1000*VAL E$
2000 CLS
2005 PRINT "ENTER ACCOUNT NAME:-
"
2010 INPUT E$
2020 IF LEN E$>20 THEN LET E$=E$
( TO 20)
2030 PRINT ,,"██ ";E$
2040 PRINT ,,"ENTER AMOUNT. ""-"
 =PURCHASES
+""=RETURNS"
2050 INPUT A
2060 PRINT "██ £";A
2070 PRINT ,,"ENTER REFERENCE CO
DE"
2080 INPUT R$
2090 IF LEN R$>6 THEN LET R$=R$(
 TO 6)
2100 PRINT "██ REF: ";R$
2110 PRINT ,,"V.A.T. INCLUSIVE O
R EXCLUSIVE? PRESS ""I"" OR ""E
"". "
2120 IF INKEY$<>"" THEN GOTO 212
0
2130 LET F$=INKEY$
2140 IF F$<>"I" AND F$<>"E" THEN
 GOTO 2130
2150 PRINT ,,"V.A.T. ";"INCLUSIV
E" AND F$="I";"EXCLUSIVE" AND F$
="E"
2160 PRINT ,,"IS ENTRY CORRECT?
(YES OR NO)?"
2170 IF INKEY$<>"" THEN GOTO 217
0
2180 LET G$=INKEY$
2185 IF G$="N" THEN CLS
2190 IF G$="N" THEN GOTO 2410
2200 IF G$="Y" THEN GOTO 2220
2210 GOTO 2180
2220 FOR X=1 TO NA
2230 IF A$(X, TO LEN E$)=E$ THEN
 GOTO 2280
2240 NEXT X
2250 LET NA=NA+1
2260 IF NA>50 THEN GOTO 2470
2265 LET X=NA
2270 LET A$(X)=E$
2280 LET AC=AC+1
2285 IF AC>200 THEN GOTO 2470
2290 LET B$(AC,1)=CHR$ X
2300 LET B$(AC,2 TO 9)=D$(2)
2310 LET B$(AC,10 TO 20)=STR$ A
2320 LET B$(AC,21 TO 26)=R$
2330 LET B$(AC,27)=F$
2340 CLS
2350 PRINT "ACCOUNT ENTERED."
2360 PRINT ,,A$(CODE B$(AC))
2370 PRINT ,,B$(AC,2 TO 9),"£";B
$(AC,10 TO 20)
2380 PRINT ,,"REF:";B$(AC,21 TO
26)
2385 LET V=0
2390 PRINT "VAT. £";
2400 IF B$(AC,27)="I" THEN LET V
=VAL B$(AC,10 TO 20)/100*15
2405 PRINT V
2410 PRINT ,,"OPTIONS:-"
2420 PRINT ,,"1. ENTER ANOTHER A
CCOUNT.","2. RETURN TO MENU."
2430 IF INKEY$<>"" THEN GOTO 243
0
2440 IF INKEY$="" THEN GOTO 2440
2450 IF INKEY$="1" THEN GOTO 200
0
2460 IF INKEY$="2" THEN GOTO 105
0
2470 GOTO 2430
2475 CLS
2480 PRINT "OUT OF ROOM, PLEASE
SAVE THIS FILE AND START A FRE
SH FILE FOR NEW ACCOUNTS. BALANC
E THIS FILE TO MAKE ROOM."
2490 PRINT ,,"PRESS ANY KEY FOR
MENU."
2500 IF INKEY$<>"" THEN GOTO 250
0
2510 IF INKEY$="" THEN GOTO 2510
2520 GOTO 1050
3000 CLS
3005 LPRINT "ACCOUNTS STANDING A
T ";D$(2)
3006 LPRINT
3010 FOR X=1 TO NA
3020 LPRINT X;" ";A$(X)
3030 NEXT X
3040 GOTO 1050
4000 CLS
4010 PRINT "ENTER ACCOUNT TO BE
DELETED."
4020 INPUT E$
4025 IF LEN E$>20 THEN LET E$=E$
( TO 20)
4030 PRINT ,,"██ ";E$
4040 PRINT ,,"PRESS ""D"" TO DEL
ETE ACCOUNT      ANY OTHER KEY RE
TURNS TO MENU."
4050 IF INKEY$<>"" THEN GOTO 405
```

```
0
4060 IF INKEY$="" THEN GOTO 4060
4070 IF INKEY$<>"D" THEN GOTO 1050
4080 FAST
4090 FOR X=1 TO NA
4100 IF A$(X, TO LEN E$)=E$ THEN GOTO 4120
4110 NEXT X
4112 SLOW
4113 PRINT ,,"ACCOUNT NOT FOUND."
4114 GOTO 4210
4120 FOR Y=X TO NA-1
4130 LET A$(Y)=A$(Y+1)
4140 NEXT Y
4150 LET A$(NA)=""
4160 LET NA=NA-1
4170 FOR Y=1 TO AC
4180 IF CODE B$(Y)=X THEN GOTO 4300
4190 NEXT Y
4200 PRINT ,,"ACCOUNT DELETED."
4210 PRINT ,,"PRESS ANY KEY TO RETURN TO MENU."
4215 SLOW
4220 IF INKEY$<>"" THEN GOTO 4220
4230 IF INKEY$="" THEN GOTO 4230
4240 GOTO 1050
4300 FOR Z=Y TO AC-1
4310 LET B$(Z)=B$(Z+1)
4320 NEXT Z
4330 LET B$(AC)=""
4340 LET AC=AC-1
4350 GOTO 4190
5000 CLS
5010 PRINT "ENTER ACCOUNT NAME."
5015 INPUT E$
5016 IF LEN E$>20 THEN LET E$=E$( TO 20)
5020 FAST
5030 FOR X=1 TO NA
5040 IF A$(X, TO LEN E$)=E$ THEN GOTO 5100
5050 NEXT X
5060 PRINT ,,"ACCOUNT NOT FOUND."
5070 GOTO 5220
5100 LPRINT "ACCOUNT ";X
5102 LPRINT "████████████████████████████████"
5103 LPRINT A$(X)
5104 LPRINT "BALANCE B/F £";B(X,1)
5105 LET TOT=B(X,1)
5110 FOR Y=1 TO AC
5115 LET V=0
5120 IF CODE B$(Y)<>X THEN GOTO 5200
5130 LPRINT B$(Y,2 TO 9),"REF:";B$(Y,21 TO 26)
5140 LET A=VAL B$(Y,10 TO 20)
5145 LET TOT=TOT+A
5150 IF B$(Y,27)="I" THEN LET V=A/100*15
5160 LET A=A-V
5170 LPRINT "£";A;" INCL. £";V;" VAT."
5200 NEXT Y
5205 LPRINT
5206 LPRINT "TOTAL £";TOT
5210 LPRINT "████████████████████████████████"
5220 SLOW
5230 PRINT "PRESS ANY KEY TO RETURN TO MENU"
5240 IF INKEY$<>"" THEN GOTO 5240
5250 IF INKEY$="" THEN GOTO 5250
5260 GOTO 1050
6000 FAST
6010 CLS
6015 LET TOT=0
6016 LET TV=0
6017 LET TR=0
6018 LET TVR=0
6020 FOR X=1 TO NA
6030 LET B(X,1)=B(X,1)+B(X,2)+B(X,3)
6040 LET B(X,2)=0
6050 LET B(X,3)=0
6060 NEXT X
6070 FOR X=1 TO AC
6080 LET Y=CODE B$(X)
6090 LET A=VAL B$(X,10 TO 20)
6100 LET V=0
6110 IF B$(X,27)="I" THEN LET V=A/100*15
6120 LET A=A-V
6130 LET B(Y,2)=B(Y,2)+V
6140 LET B(Y,3)=B(Y,3)+A
6150 IF SGN A=1 THEN LET TOT=TOT+A
6160 IF SGN A=1 THEN LET TV=TV+V
6165 IF SGN A=-1 THEN LET TR=TR+A
6166 IF SGN A=-1 THEN LET TVR=TVR+V
6170 NEXT X
6175 LPRINT "BALANCES FOR PERIOD FROM"
6176 LPRINT D$(1);" TO ";D$(2)
6177 LPRINT "████████████████████████████████"
6180 FOR X=1 TO NA
6190 LPRINT A$(X)
6200 LPRINT "£";B(X,2)+B(X,3);" INC. £";B(X,2);" VAT"
6210 NEXT X
6220 LPRINT
6230 LPRINT "CARRY DOWN TO NOMINAL LEDGER:-"
6240 LPRINT "RETURNS:","£";TOT
6250 LPRINT "RETURNS VAT:","£";TV
6260 LPRINT "PURCHASES:","£";-TR
6270 LPRINT "PURCHASES VAT:","£";-TVR
6280 LPRINT "████████████████████████████████"
6281 FOR X=1 TO AC
6282 LET B$(X)=""
6283 NEXT X
6284 LET AC=0
6290 SLOW
6300 LET D$(1)=D$(2)
6310 LET D$(2)=""
6320 GOTO 1000
7000 CLS
7005 PRINT "ENTER FILE NAME:-"
7010 INPUT F$
7020 PRINT ,,"██ ";F$
7030 PRINT ,,"START TAPE RECORDER AND PRESS ANY KEY."
7040 IF INKEY$<>"" THEN GOTO 7040
7050 IF INKEY$="" THEN GOTO 7050
7060 SAVE F$
7070 GOTO 1000
```

To convert the listing of the Purchase ledger to that of a Sales ledger, simply replace the revelant lines in the above listing to read:

```
1070 PRINT ,,"SALES LEDGER OPTIONS:-",,"████████████████████████████████",,"1. MAKE ENTRY."
2070 PRINT ,,"ENTER AMOUNT. ""+"" =SALES -""=RETURNS"
6240 LPRINT "SALES:","£";TOT
6250 LPRINT "SALES VAT:","£";TV
6260 LPRINT "RETURNS:","£";-TR
6270 LPRINT "RETURNS VAT:","£";-TVR
```

ZX81 GAMES

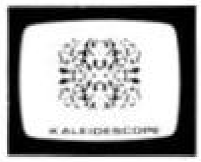
from

J.K. GREYE SOFTWARE LTD

THE NEW GENERATION SOFTWARE HOUSE

"Without question the finest machine code games available today"......... J.N. ROWLAND Product Manager for W.H. SMITH.

GAMESTAPE 1 for 1K ______________ **only £3.95**
10 Games incl. ASTEROIDS, UFO, CODE, BOMBER, GUILLOTINE, KALEIDESCOPE, etc.
PROBABLY THE BEST VALUE 1k TAPE AVAILABLE.



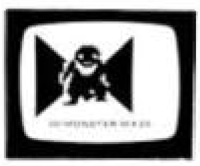
We've done in 1k, games which some of our competitors require 16k to do!



GAMESTAPE 2 for 16K ______________ **only £3.95**
***STARFIGHTER** Superb machine code Space Battle. Set against a background of twinkling stars, with stunning explosions — if you can hit the enemy!
PYRAMID Can you move the Pyramid? Make a mistake and it will collapse! A Thinkers game.
ARTIST The ultimate Graphic Designers aid. 8 Directions, 10 Memories, SAVE, COPY, RUBOUT, CLS, etc.

GAMESTAPE 3 for 16K ______________ **only £4.95**
***CATACOMBS** A Multi-Level Graphics Adventure. Each level can contain up to 9 Rooms, 8 Passages, 7 Monsters, Food, Gold, Traps, Phantoms, an Exit (to the next level), and there's an infinite number of levels.
NOTE. . . . This is NOT one of the necessarily limited text Adventures as sold elsewhere.
"An excellent addictive game which will keep you amused for hours". . . . *COMPUTER & VIDEO GAMES.*

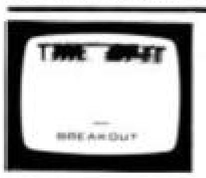




GAMESTAPE 4 for 16K ______________ **only £4.95**
***3D MONSTER MAZE** The Game to Top All Others. Unbelievable Graphics! Can you find your way through the Maze? The EXIT is there somewhere, but then so is a T.REX, and its after YOU! All in 3D (the T.REX will actually run towards you in full perspective!), you've never seen anything like this before!
"3D MONSTER MAZE is the best game I have seen for the ZX81". . . COMPUTER & VIDEO GAMES
"If I had to choose just one programme to impress an audience with the capabilities of the ZX81, then J.K. Greye's 3D MONSTER MAZE would be the one without doubt" . . . . ZX COMPUTING. *"Brilliant, brilliant, brilliant!"*. . . . *POPULAR COMPUTING WEEKLY*

GAMESTAPE 5 for 16K ______________ **only £4.95**
***3D DEFENDER** The Ultimate Space Game. Super fast Machine Code 3D version of the Arcade favourite. You have to save your home planet from the marauding Alien Spacecraft. This is all in 3D, your viewscreen shows you the view out of your fighters cockpit window. The backdrop moves when you turn, or fly up or down (8 flight directions), just as if you were really flying it! But then YOU ARE! The Enemy Saucers will actually zoom towards you in 3D, and shoot you if you let them! Your display includes Score, Shield Strength, Altitude, Proximity, Forward Radar and your viewscreen, which shows your rotating home planet, backdrop of Stars, Meteors, Explosions, Plasma Blasts, your Photon Beams, up to 4 Enemy Saucers and of course its all in full 3D!
"Another 3D winner". . . . . *SINCLAIR USER*





GAMESTAPE 6 for 1K ______________ **only £1.95**
***BREAKOUT** Super Fast Full Screen Display Game. Your all time favourite with an added twist. See how much Money you can win and watch the pounds convert to Dollars. All in Machine Code for Fast Action with 3 Speeds, 2 Bat Sizes and three angles of rebound! The best BREAKOUT around and at this price you can't go wrong!
"The best of its kind" . . . . *WHICH MICRO & SOFTWARE REVIEW*

GAMES MARKED * INCL. MACHINE CODE.
Prices include VAT and U.K. P. & P.
(Add appropriate Postage on Foreign Orders). Cheques/P.O.s to

J.K. GREYE SOFTWARE LTD

Dept ZX16, Brendan Close, Oldland Common, Bristol BS15 6QE

CREDIT CARD SALES: FOR INSTANT DESPATCH, **BY PHONE ONLY**
TEL: 01-930-9232 (9 am - 7 pm)

If you prefer to see before buying, our range of **GAMESTAPES** are stocked by the following stores.

BUFFER MICROSHOP ______ 374A Streatham High Rd., London SW16;
GAMER ______ 24 Gloucester Rd., Brighton;
GEORGES ______ 89 Park St., Bristol, Avon;
MICROSTYLE ______ 29 Belvedere, Lansdown Rd., Bath, Avon;
MICROWARE ______ 131 Melton Rd., Leicester;
SCREEN SCENE ______ 144 St. Georges Rd., Cheltenham, Glos;
W.H. SMITH ______ Over 200 Computer Branches;
ZEDXTRA ______ 5 School Lane, Kinson, Bournemouth, Dorset.

TRADE & EXPORT ENQUIRIES WELCOME

HOW DARE THEY!

If you see an advertisement in the press, in print, on posters or a cinema commercial which makes you angry, write to us at the address below. (TV and radio commercials are dealt with by the I.B.A.)

The Advertising Standards Authority.
If an advertisement is wrong, we're here to put it right.

ASA Ltd, Brook House, Torrington Place, London WC1E 7HN.

• READERS •

Reach over 70,000 **ZX** Enthusiasts. Swop, sell or buy your **ZX** Hardware, Software, Peripherals etc. in our
NEW CLASSIFIED SECTION

STARTING SOON IN

ZX COMPUTING

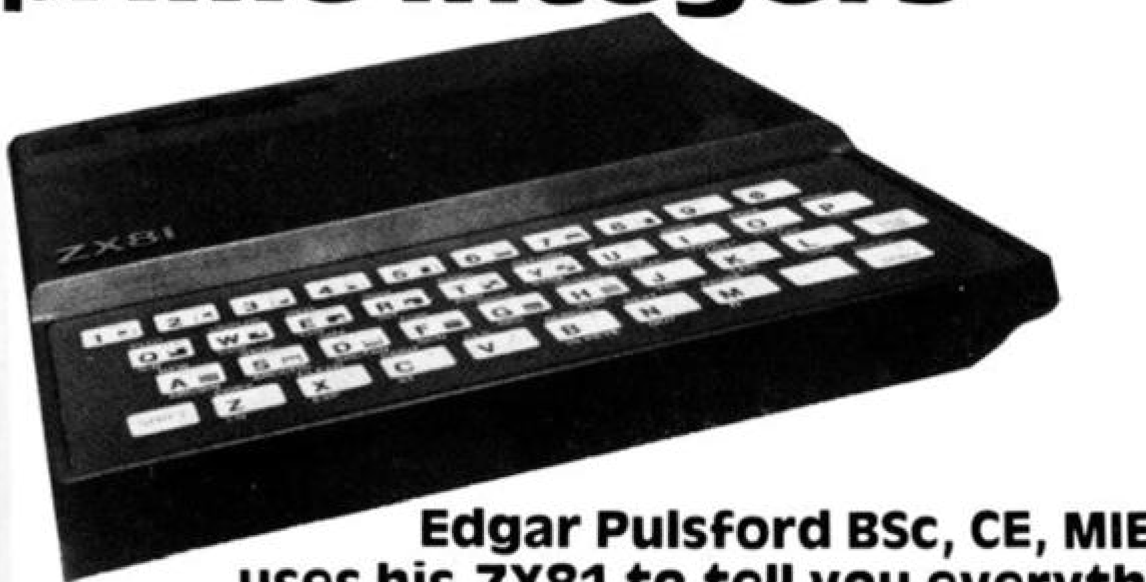
Britain's Biggest Magazine for the Sinclair User.

PHONE
BRIDGETTE ON

01-437 1002

FOR DETAILS

On the density of prime integers



Edgar Pulsford BSc, CE, MIERE, uses his ZX81 to tell you everything you ever wanted to know about prime integers . . . but were afraid to ask.

A computer is programmed to identify and count the primes in a range of consecutive integers centred on a power of 10, eg from 9,500 to 10,500. The proportion of primes in the range is calculated for each decade from 10 to 10^9.

If the proportion of primes in the integers of the range is termed the *density* of primes, then it is found that:

The product of the density of primes and the natural logarithm of the central integer of the range appears to be unity.

or:

The mean spacing of primes about N equals $\log_e N$.

These results have irregularities because of the non-uniform spacing of the primes. Also, the number of integers in the range is small compared to the magnitude of the central integer, and yet large enough to reduce statistical errors to minor importance.

All integers are positive, and primes are those with no factors other than themselves and unity. The principal symbols involved with the density of prime integers are given below.

X — Lower bound of range.
Y — Upper bound of range.
W — Width of range = (X − Y) consecutive integers.
N — Central integer of the range = (X + Y)/2.
P — Number of primes in the range.
R — Proportion (or density) of primes = P/W.
ln N — Natural logarithm of N.

The empirical result in symbols can be seen in the equation below.

$R \times \ln N = 1$

Counting of primes

This work was prompted by a program published in *ZX Computing*, Vol. 1, No. 1, page 29, which program was modified to give a serial number to each prime. This proved to be the important factor for what follows resulting in the primes being considered as a definite class of integers, little notice being taken of their values. The study became one of the density of primes in the integers, particularly with relation to the position in the integers where the density was measured.

The modified program referred to is given in Fig. 1. Figure 2 gives the result of printing out every hundredth prime, and also shows the ratio of the prime to its serial number. Figure 3 shows a graph of this ratio against its serial number. The shape of this curve has not been studied, but the fact that it was a smooth curve indicated that there was some relationship worthy of further investigation.

It was then decided to study small regions of the integers, finding the proportion of primes in each region, and relating this to the magnitude of the central integer. For example, in the region 500 to 1,500, 144 primes were found. The proportion is therefore 144/1,000 or 0.144, which was related to 1,000, the central integer between 500 and 1,000. A measurement of this kind was made, using the program of Fig. 4, at each decade from 10 to 10^9, which is about as far as the ZX81 will conveniently go.

The principles involved are:
(a) Integers of the series $6M \pm 1$ are chosen for testing. It may be seen that this series contains all the primes, although all numbers of this series are not primes. Seeing that $6M \pm 1$ and $6M \pm 3$ represent all the odd integers (but $6M \pm 3$ are composite, being divisible by 3) it follows that $6M \pm 1$ contains all the integers which might be primes.

The program starts with a trial integer of the form 6M − 1 near the lower bound of the region of interest, tests it, adds two, tests the number so formed, then adds four and tests that, and so on, counting the primes as they are found.
(b) The divisors (D) used for the tests start at three, and each time a new divisor is needed, two is added to the D just used so that all the odd integers are used in turn. There is no point in using even integers.
(c) The composite nature of the number under test is detected by:

550 IF N/D − INT(N/D) = 0, etc

If this is true then N/D is an integer and equal to INT (N/D). Successive values of D are tried and if the above is found to be true, then the next N is formed and the testing resumed.
(d) The primality of the tested number is detected by limiting the value of D to a maximum equal to the square root of N. If no divisor is found up to this value, then N is prime. Steps are taken to count and record (if desired) the prime so found, and then the next number for trial is formed and tested for primality or compositeness.

A program devised to do this is given in Fig. 4. The work stops when N is greater than the upper bound of the chosen region. Typical results are printed out in Fig. 4a and fuller results are tabulated in Table 1.

A new region is then chosen, typically one decade greater, and the work continued. Because of the slowness of the testing when N is large, say greater than 10^6, it is convenient to reduce the number of primes counted at any particular decade, and to accept a worsening of statistical accuracy.

Treatment of results

As the observations were made on regions one decade apart, it seemed convenient to plot (manually) R against $\log_{10} N$. Figure 5 is the result. Although at first, the base 10 was used, later it became apparent that the natural base, e, was to be preferred. The plot of R against ln N is therefore shown.

The shape of this curve prompted examination to see if it were an exponential or a rectangular hyperbola. It was soon found that it was a good fit to the equation:

X × Y is a near constant.

If the product of R and ln N are formed, it is seen that they are not only near constant, but that constant is near unity. Thus, the empirical relation:

$R \times \ln N = 1$ (approximately)

is obtained, where N extends over the range from 10 to 10^9.

This result has the quality of being mathematically simple, which is satisfying. But it is possible to go a stage further. R × ln N may be written:

$\ln N^R$ or $\ln N^{P/W}$

and either of these equivalent expressions may be equated to unity (approx).

And since the number whose ▶

natural logarithm is unity is e, or exp, it is possible to write:

$N^{P/W} = e$

where e is equal to 2.718

Statement of the result

If there are P primes in a region W of the natural integers of which the central integer is N, then, if R = P/W,

$R \times \ln N = 1$ or $N^{P/W} = e$

The accuracy and 'constancy' of R × ln N is shown in Table 1. The effects of the statistics of sampling have been taken into account only insofar as a reasonable sample size for P has been aimed at, consistent with the time available for the work.

Testing the relationship

Having made the observations and developed the relationship, it remains to test it to see if it can be used to forecast new information.

A test program was written in which a series of Ns from 3.2×10^2 to 3.2×10^8 were chosen. These may be thought of as the half-points of decades because log 3.2 is about 0.5. For each of these Ns, the value of W which was expected to contain 75 primes was calculated and the computer set to count the actual number in this W, the result being compared to the expectation. The program and results are given in Fig. 6 and Table 2 (test results). It will be seen that there is a good agreement between measurement and expectation.

Material for tests is also found in the results of the first experiment, Figs. 1 and 2. The empirical relationship can be used to calculate the number of primes expected and to compare the results with 100. This was done and is shown in the results of the program in Fig. 7. The results are fairly consistant with expectations, but this is only a small range over which to do the test. The figures are included for interest; their mean is 100.8. It is considered that they support the empirical relationship.

Various observations

The program of Fig. 4 is so designed that it is essential that the first integer in the range W offered for test is of the form 6M − 1. To ensure this, when the value X of the lower bound has been found, the starting integer for the test, X', is found using:

$X' = 6 \times INT(X/6) - 1$

INT (X/6) gives the value of M, and then the expression gives an integer which is a few units only away from the value of the lower bound, X.

The value of the upper bound, Y, is used without modification for ending the search.

For the larger Ns the computing takes a long time, even in Fast mode. Reducing the number of primes has been used to shorten the work but continuous running day and night has been neccessary. If the scope of this work is to be extended, a faster working and larger number-handling computer will be needed.

The region about 10^7 produced anomalies, suggesting that there is a scarcity of primes there. This needs looking into. It is known that there are regions where there are long successions of composites (see text books on Theory of Numbers)

N	W	P	R=P/W	ln N	R×ln N	$N^{P/W}$ e = 2.718
10	20	9	0.45	2.303	1.04	2.82
10^2	106	22	0.2075	4.605	.956	2.60
10^3	1006	145	0.1441	6.908	.996	2.706
10^4	1006	109	0.1083	9.210	.998	2.711
10^5	1006	87	0.0865	11.513	.996	2.707
10^6	1006	72	0.0716	13.816	.989	2.69
10^7	1006	52	0.0516	16.118	.833	2.30
10^7	2000	114	0.057	16.118	.919	2.51
10^7	2624	160	0.061	16.118	.983	2.67
(9×10^6)	998	59	0.0591	16.013	.947	2.516
10^8	202	11	0.0544	18.42	1.003	2.727
10^9	202	9	0.0446	20.72	.923	2.517
	512	25	0.048	20.72	.995	2.704

Table 1. The full results from the program given in Fig. 4.



```
N = 320
P = 75
W = 432
LIMITS 104 TO 536
 SEARCH FROM 101 TO 536

AT N = 320 THERE ARE 71 PRIMES W
HERE 75 WERE EXPECTED

*****        *****        *****        **

N = 3200
P = 75
W = 605
LIMITS 2898 TO 3502
 SEARCH FROM 2897 TO 3503

AT N = 3200 THERE ARE 71 PRIMES
WHERE 75 WERE EXPECTED

*****        *****        *****        **
N = 32000
P = 75
W = 778
LIMITS 31611 TO 32389
 SEARCH FROM 31607 TO 32389

AT N = 32000 THERE ARE 73 PRIMES
 WHERE 75 WERE EXPECTED
*****        *****        *****        **
N = 320000
P = 75
W = 950
LIMITS 319525 TO 320475
 SEARCH FROM 319523 TO 320475
```

Table 2. The results obtained from the program given in Fig. 6.

but this region has not yet been examined. Extending W, or shifting a little way from 10^7 gives results more in line with the general findings of this work. For completeness several results in this region are included in Table 3.

The number of *significant* figures retained in the tables is thought to be not inconsistant with the experimental results. Programming devices were used to reduce many results to three significant figures.

A machine plot of the results of the first experiment was made, but because of the coarseness of PLOT it shows irregularities which are absent from the manual plot.

It is known that primes sometimes occur in pairs, both 6M − 1 and 6M + 1 being prime. Examples are 11,13; 17,19; 29,31; etc. These pairs were noticed at all stages of the present work. It is known that pairs become rarer as N becomes larger, so a program was devised to count pairs at various decades of N. Beyond supporting this statement, no other information about them was noticed, but the density of prime pairs might make the subject of a study arising out of the work described here.

It follows that ln N is the mean spacing of primes in the vicinity of N.

Conclusion

It is thought that new light may have been shed on the subject of the density of primes in the set of ordered integers. So far no reference has been found to work of this kind in the few books I have read. However, there is a reference in **Numbers**, by L. F. Taylor (Faber) p. 81, to work done in 1896, which gave the number of primes (Px) less than a given integer x, as in x/ln X as x tends to infinity, but this does not seem the same as the result here described.

Further developments must be left to more skilled mathemeticians and computer programmers than the present author is ever likely to be.

Bibliography

H. S. Hall and S. R. Knight: **Higher Algebra**, Macmillan, 4th edition, 1927. Chapter 30: Theory of Numbers.

Tobias Dantzig: **Number, the Language of Science,** Allen and Unwin, 1942. Chapter 3: sections on prime numbers.

F. S. Merritt: **Mathematics Manual**, McGraw-Hill, 1962. Chapter 11: Numerical Integration.

L. F. Taylor: **Numbers,** Faber and Faber, 1970. Chapters 7 and 8: Prime Numbers.

```
AT N = 320000 THERE ARE 75 PRIME
S WHERE 75 WERE EXPECTED

*****          *****          *****          **
N = 3200000
P = 75
W = 1123
LIMITS 3199438 TO 3200561
 SEARCH FROM 3199433 TO 3200562

AT N = 3200000 THERE ARE 85 PRIM
ES WHERE 75 WERE EXPECTED

*****          *****          *****          **
N = 32000000
P = 75
W = 1296
LIMITS 31999352 TO 32000648
 SEARCH FROM 31999349 TO 3200064
8

AT N = 32000000 THERE ARE 73 PRI
MES WHERE 75 WERE EXPECTED

*****          *****          *****          **

N = 320000000
P = 75
W = 1468
LIMITS 319999270 TO 320000730
 SEARCH FROM 319999270 TO 320000
730

AT N = 320000000 THERE ARE 73 PR
IMES WHERE 75 WERE EXPECTED

*****          *****          *****          **
```

N	Search Limits	Expected	Counted
3.2×10^2	29 608	100	95
3.2×10^3	2795 3604	100	93
3.2×10^4	31481 32519	100	99
3.2×10^5	319381 320634	100	101
3.2×10^6	3199247 3200749	100	107
3.2×10^7	31999565 32000432	50	51
3.2×10^8	319999510 320000490	50	49

Table 3. The results obtained when N is large and W is shifted a little way from 10^7.

```
  10 REM "PP1"
  15 REM "E.W.P. 16/8/82 (REF.
"ZX81.,COMPUTING" VOL 1 NO.1,
PAGE 29" (MODIFIED))
  20 REM "THIS PROGRAM PRINTS
EVERY 100TH PRIME WITH ITS
SERIAL NUMBER AND THE RATIO P/S"
  30 LPRINT "SERIAL";TAB 8;"PRIM
E";TAB 15;"RATIO P/S"
  40 LPRINT
  50 FAST
  60 DIM Q(2000)
  70 LET Z=1
  80 LET Q(1)=2
  90 FOR G=3 TO 30000 STEP 2
 100 FOR H=1 TO Z
 110 IF INT (G/Q(H))*Q(H)=G THEN
 GOTO 180
 120 NEXT H
 130 LET Z=Z+1
 140 LET Q(Z)=G
 150 FOR C=1 TO 35
 160 IF C=H/100 THEN LPRINT TAB
2;H;TAB 8;G;TAB 15;.01*INT ((G*1
00)/H+.5)
 170 NEXT C
 180 NEXT G
 200 SAVE "FF█"
 210 GOTO 30
```

Fig. 1. The modified program to give a serial number to each prime.

SERIAL	PRIME	RATIO P/S
100	541	5.41
200	1223	6.12
300	1987	6.62
400	2741	6.85
500	3571	7.14
600	4409	7.35
700	5279	7.54
800	6133	7.67
900	6997	7.77
1000	7919	7.92
1100	8831	8.03
1200	9733	8.11
1300	10657	8.2
1400	11657	8.33
1500	12553	8.37
1600	13499	8.44
1700	14519	8.54
1800	15401	8.56
1900	16381	8.62
2000	17389	8.69

Fig. 2. The result of printing out every hundredth prime and the ratio, R.

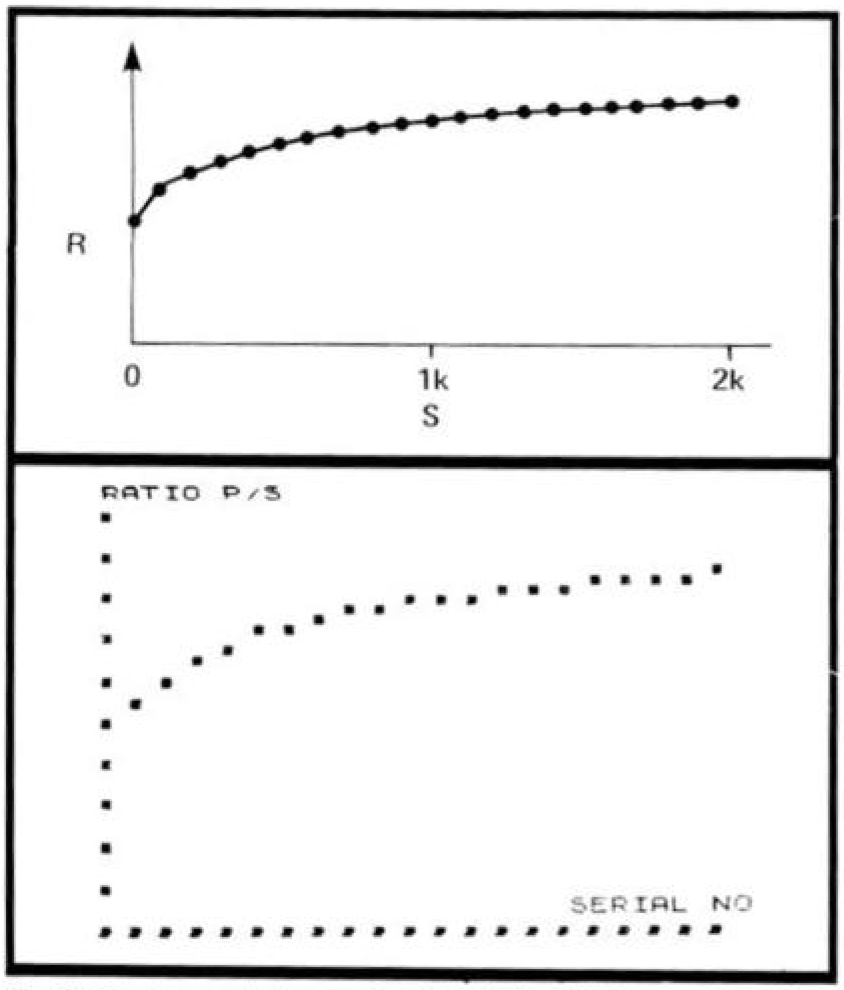


Fig. 3. The top graph shows what the relationship between R and its serial number should be. The bottom graph shows the actual printout.

```
  10 REM "PP4"
  20 PRINT "**THIS PROGRAM GIVES
 PRIMES AND SERIAL NUMBERS BETWE
EN SPECIFIED LIMITS**"
  25 PRINT "FOR LIMITS PRESS ""S
TOP"" AND ""NEWLINE"" TWICE. CON
TINUE WITH ""RUN"" AND ""NEWLINE
"""
  30 REM **LOWER LIMITS**
 100 LET A2=47
 110 LET A3=497
 120 LET A4=9497
 130 LET A5=99497
 140 LET A6=999497
 150 LET A7=9999497
 160 LET A8=99999899
 170 LET A9=999999899
 180 REM **UPPER LIMITS**
 200 LET U2=153
 210 LET U3=1503
 220 LET U4=10503
 230 LET U5=100503
 240 LET U6=1000503
 250 LET U7=10000503
 260 LET U8=100000101
 270 LET U9=1000000101
 280 PRINT "INPUT LOWER LIMIT AS
""AN"""
 290 INPUT AN
 400 LET N=AN
 410 PRINT "INPUT UPPER LIMIT AS
""UN"""
 420 INPUT UN
 430 LET U=UN
 440 PRINT N;" TO ";U;" DIFF.= "
;U-N
 450 LET S=1
 470 PRINT "PRESS C TO PROCEED.
PROGRESS MAY BE SEEN BY PRESSING
""BREAK"".      CONTINUE WITH C A
ND NEWLINE"
 480 IF INKEY$="" THEN GOTO 480
 490 IF INKEY$="C" THEN GOTO 500
 500 CLS
 510 FAST
 520 LET C=1
 530 LET D=3
 540 SCROLL
 550 IF N/D-INT (N/D)=0 THEN GOT
O 620
 560 IF D>SQR N THEN GOTO 600
 570 LET D=D+2
 580 GOTO 550
 600 PRINT TAB 5;S;TAB 12;N
 610 LET S=S+1
 620 LET C=C+1
 630 IF C>2 THEN GOTO 660
 640 LET N=N+2
 650 GOTO 530
 660 SCROLL
 670 PRINT TAB 5;"*"
 680 LET N=N+4
 690 IF N>U THEN GOTO 705
 700 GOTO 510
 705 SLOW
 710 SCROLL
 720 PRINT "CENTRE OF RANGE, N =
";(AN+UN)/2
 730 SCROLL
 740 PRINT "RANGE, W =";(UN-AN)
 750 SCROLL
 760 PRINT "NO. OF PRIMES. P =";
S-1
 770 SCROLL
 780 PRINT "PROP'N OF PRIMES,R =
";(S-1)/(UN-AN)
 790 SCROLL
 800 PRINT "NAT'L LOG N,LN N =";
LN ((AN+UN)/2)
 810 SCROLL
 820 PRINT "PRODUCT, R X LN N ="
; ((S-1)/(UN-AN))*LN ((AN+UN)/2)
 830 SCROLL
 835 SCROLL
 840 PRINT "TABULATE IN WRITING"
 845 SCROLL
 846 SCROLL
 850 PRINT "PRESS C TO CONTINUE,
 860 SCROLL
 870 PRINT """BREAK"" THEN ""NEW
LINE"" FOR PROG."
 880 IF INKEY$="" THEN GOTO 880
 890 IF INKEY$="C" THEN CLS
 900 GOTO 20
1000 SAVE "PP■"
1010 GOTO 20
```

Fig. 4. A program to find the proportion of primes in a particular region, relating the result to the magnitude of the central integer.

```
    *
    108       10499
    109       10501
    *
CENTRE OF RANGE, N =10000
RANGE, W =1006
NO. OF PRIMES. P =109
PROP'N OF PRIMES,R =0.1083499
NAT'L LOG N,LN N =9.2103404
PRODUCT, R X LN N =0.99793946

TABULATE IN WRITING

PRESS C TO CONTINUE, OR
"BREAK" THEN "NEWLINE" FOR PROG.
    *

    59        9000499
    *
CENTRE OF RANGE, N =9000000
RANGE, W =998
NO. OF PRIMES. P =59
PROP'N OF PRIMES,R =.059118237
NAT'L LOG N,LN N =16.012735
PRODUCT, R X LN N =0.94664466

TABULATE IN WRITING
PRESS C TO CONTINUE, OR
"BREAK" THEN "NEWLINE" FOR PROG.
```

Fig. 4a. Typical results from the program given in Fig. 4.

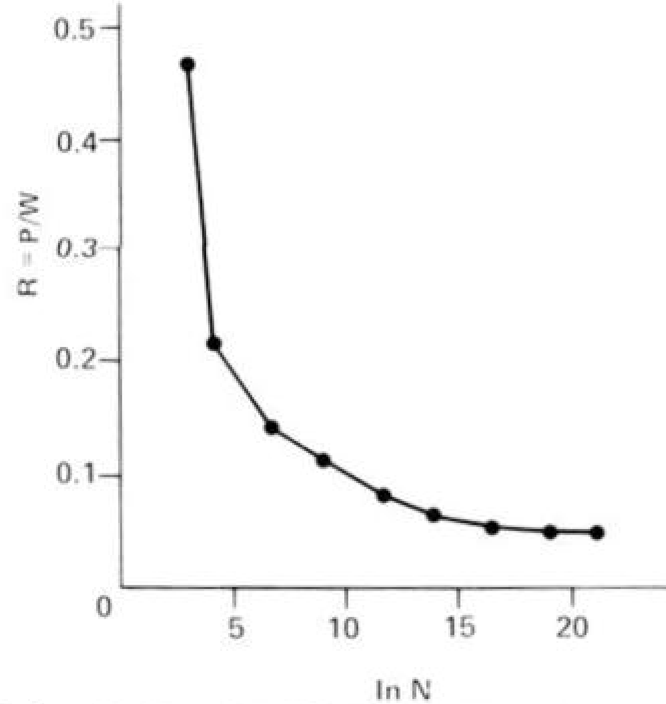


Fig. 5. A graphic illustration of R plotted against ln N.

```
  10 REM "PTP"
 110 LET N=320
 115 PRINT "N = ";N
 116 LPRINT "N = ";N
 120 LET P=75
 125 PRINT "P = ";P
 126 LPRINT "P = ";P
 130 LET W=INT (P*LN N)
 140 PRINT "W = ";W
 141 LPRINT "W = ";W
 150 PRINT "LIMITS ";N-INT (0.5*
(P*LN N));" TO ";N+INT (0.5*(P*L
N N))
 151 LPRINT "LIMITS ";N-INT (0.5
*(P*LN N));" TO ";N+INT (0.5*(P*
LN N))
 160 LET A=N-INT (0.5*(P*LN N))
 170 LET X=6*INT (A/6)-1
 190 LET Y=A+W
 200 PRINT " SEARCH FROM ";X;" T
O ";Y
 201 LPRINT " SEARCH FROM ";X;"
 202■PRINT
 203 LPRINT
 204 GOTO 225
 210 PRINT "PRESS C TO CONTINUE"
 220 IF INKEY$<>"C" THEN GOTO 22
0
 225 PAUSE 1000
 230 CLS
 235 FAST
 240 SCROLL
 250 LET S=1
 260 LET C=1
 270 LET D=3
 290 IF X/D-INT (X/D)=0 THEN GOT
O 370
 300 IF D>SQR X THEN GOTO 330
 310 LET D=D+2
 320 GOTO 290
 330 PRINT TAB 5;S;TAB 12;X
 335 SCROLL
 340 LET S=S+1
 350 LET C=C+1
 360■IF C>2 THEN GOTO 400
 370 LET X=X+2
 380 GOTO 270
 400 LET X=X+4
 410 IF X>Y THEN GOTO 430
 420 GOTO 260
 430 CLS
 435 SLOW
 440 PRINT "AT N = ";N;" THERE A
RE ";S;" PRIMES WHERE ";P;" WERE
 EXPECTED"
 441 LPRINT "AT N = ";N;" THERE
ARE ";S;" PRIMES WHERE ";P;" WER
E EXPECTED"
 442 LPRINT
 443 LPRINT "*****           *****
*****        **"
 444 LPRINT
 450 LET N=10*N
 460 PRINT
 470 GOTO 115
 999 STOP
1000■SAVE "PT■"
1010 GOTO 110
```

Fig. 6. A program to calculate the value of W from a series of Ns. If you want a break between decade tests, simply omit lines 204 and 225.

```
   5 REM "PP3"
   7■PRINT "**INPUT THE PRIMES
OF ""PP1"" IN ORDER. THESE ARE
100 PRIMES     APART**"
   8 PRINT "*NOTE-THE CALCULATED
 NO.IS NEAR 100 EACH TIME**"
   9 PRINT
  10 INPUT A
  20 INPUT B
  25 LET C=B-A
  30 PRINT A,,,
  35 LET D=(A+B)/2
  50 PRINT INT ((C/LN D)+.5)
  70 LET A=B
  80 GOTO 20
 100 SAVE "PP■"
 110 GOTO 5
```

```
**INPUT THE PRIMES  OF  "PP1" IN
 ORDER. THESE ARE 100 PRIMES
APART**
*NOTE-THE CALCULATED NO.IS NEAR
100 EACH TIME**

541
                101
1223
                104
1987
                97
2741
                103
3571
                101
4409
                103
5279
                99
6133
                98
6997
                103
7919
                101
8831
                99
9733
                100
10657
                107
11657
                95
12553
                100
13499
                107
14519
                92
15401
                101
16381
                104
17389
```

Fig. 7. Using the results of the first experiment in Figs. 1 and 2, the empirical relationship can be used to calculate the number of primes expected and to compare the results with 100.

What can I do with 1K



You may think 1K of memory is not enough to do anything useful. Roger Valentine shows you that 1K is more than enough if you program efficiently.

There has been a tendancy of late for computer memory sizes to go literally 'through the roof'. In olden days (and remember that in computer terms, the 'olden days' were only two or three years ago!), the largest possible memory size for an eight-bit microcomputer was 64K. Only a very few exclusively business machines actually achieved this maximum; the largest PET was 32K, the Apple was 48K and the Sinclair ZX80 was (theoretically) expandable up to 16K, although many ZX80 users either did not expand beyond 1K at all, or were content with the smaller 4K RAM extension. But look at the situation today – the latest 'Superpet' is available with 256K of RAM; the Sinclair Spectrum can have 48K 'on board' as standard; even the 'small' Spectrum is as large as the 'fully expanded' ZX81 (16K). And the ZX81 itself? Well, there is the 64K 'Memopack', and I believe (although I haven't actually seen this personally) there is a 128K RAM extension on the market. Sinclair's own 16K RAM Pack has been reduced in price, and any number of alternative 16K RAMs are now available.

What's in store?

If you are still working with 1K, you may well get the impression that you are becoming a 'second class citizen' in the computer world! Is there any point in even *trying* to squeeze in anything worthwhile, or should you join the rest in the Great RAM Race?

The answer, I firmly believe, is *yes*, you should persevere, and *no*, you should not expand, *until* you have learnt how to make the most of every single one of your 1,024 bytes. Let's face it, most people buy a ZX81 to *learn* about computing, and there are lessons to be learnt about *efficient* and *concise* programming which even the so-called experts have never realised. Computer programs are like gases – they tend to expand to fill the amount of space available to them (Valentine's first law!). If you have got 64K, then you will use it, *all* of it, even for the simplest of problems. And for anything complex, well, you'll just have to look around for that 128K RAM pack!

Does this matter? If larger memories *are* available, is there any point in writing efficiently? There certainly is! Apart from the obvious fact that most programmers take pride in their work (and it is difficult to be proud of something which is inefficient), consider the following situation. You have a 16K RAM Pack, and have been writing programs for months without any memory problems. One day, you decide to write a database routine, the first line of which is:

```
10 DIM A$(100, 10,15)
```

Panic! Anything you try to add produces the notorious 'error code 4', so you reduce the number of data items from 100 to 90, and then to 80. Eventually, you find you have created the world's smallest database!

Now, massive multi-dimensional arrays are *not* the sort of thing which beginners use very frequently, but look at any professionally written programs (certainly in the business field, but games and graphics programs as well), and you will see that they are extremely common. You are going to have to use them one day, and if you haven't learnt how to squeeze programs into a small amount of space by then, you are going to have problems!

Tricks of the trade

I thought it would be a good idea to gather together *all* the possible space saving techniques I could think of into one place, for permanent reference. Here, then, in no particular order, is a complete list of memory saving 'tricks' for use with your 1K ZX81.

Some of these techniques are very obvious, and are presented without explanation; others are illustrated in the sample programs. Some are mutually exclusive (if you use one, you may not be able to use another) and some appear to 'supersede' others (they achieve the same object, but more efficiently). The actual amount of memory saved, and which of the techniques is 'best', will depend on the context in which you use them.

1. Use single-letter variable names.
2. Re-use variables as often as possible.
3. Avoid using variables altogether (eg when a result is to be PRINTed, and is not required further). For example, here is a beginner's program for calculating square roots:

```
10 INPUT X
20 LET S=SQR X
30 PRINT "THE SQUARE
ROOT OF ";X;" IS "; S
```

and here is the 'improved', shortened version:

```
10 INPUT X
20 PRINT "THE SQUARE
ROOT OF ";X;" IS ";SQR X
```

4. Remember that loop-control variables take up more room than simple variables. Either avoid using FOR . . . TO loops or use the same loop control variable for every loop in the program.
5. Avoid using numbers! The ZX81 uses bytes to store a single digit number (or eight bytes for two digits, etc). Any of the following techniques may be more economical:

a) If a number is going to be used several times, assign a variable to that number.
b) Use previously assigned variables. For example,

```
10 LET A=1
20 LET B=A
30 LET C=A
```

will initialise all three variables although only one number is used.
c) Use VAL. The expression LET A=VAL "200" is more economical than LET A=200.



d) Use CODE. For numbers up to 255, this is even better; for example LET A = CODE "COS".
e) Use PI. Never forget that the ZX81 already has the value of PI in its memory. OK, so there aren't many occasions when you would want the value 3.1415927 in non-trigonometrical programs, but remember that INT PI is 3, and that some expressions (eg PRINT AT, FOR . . . TO loops, etc) treat decimal values as integers *without* requiring the word INT. Also, remember that PI/PI equals 1; PI-PI (or SIN PI, or even NOT PI, see below) equals 0, and COS PI equals −1.
f) Use Boolean logic. NOT is the easiest expression to use:

NOT (any variable with a value other than 0) = 0

and:

NOT (any variable with value 0) = 1

The benefits of using logical expressions cannot be overemphasised. Re-read chapter 10 of the ZX manual, and this time, don't skip the exercises!
6. Avoid repetitive IF statements where the condition is the same for several successive lines. This is both uneconomical and poor style. Branch to a separate routine instead, or see if the lines can be compressed into one (using Boolean logic again).
7. Keep the display file to a minimum by using fewer PRINT statements, and printing everything in the top left of the screen. Avoid using commas, which will pad out the display file with spaces.
8. Although the ZX81 will not accept multiple-statement lines, you *can* combine any number of PRINT statements (including AT and TAB) into a single line, by separating them with semicolons. (The word PRINT is only used once.)
9. Avoid using the statement STOP, by:
a) Arranging for the program to 'fall through' to a natural end at the end of the listing.
b) Making the program into a continuous loop, which can only be terminated by pressing Break.
c) Using a 'dummy variable' (unassigned) to terminate the program with report code 2.
10. Avoid using REMs (keep program notes instead!).
11. Avoid using brackets in calculations by exploiting the ZX81's expression evaluation priorities (see chapter 4 of the ZX manual).
12. Do not use low line numbers in the mistaken belief that line number 1 takes up fewer bytes than line number 1000. All line numbers occupy exactly 5 bytes. However, in the expressions GOTO and GOSUB, the numbers which follow the keyword are (to the ZX81) merely numbers, and so GOTO 1 is shorter than GOTO 1000. Therefore, arrange for all subroutines and lines which are to be frequently called by GOTO statements to have low lin numbers.
13. Use keywords in PRINT statements: "ENTER YOUR NAME" may sound better (?) than "INPUT YOUR NAME", but is less economical, because, whereas 'ENTER' must be spelt out in full, the keyword 'INPUT' may be used, and it will only occupy a single byte. (How do you set a keyword into a PRINT statement? Easy: you simply fool the computer into expecting a keyword by entering THEN (shift 3), then enter the keyword, and finally delete the word THEN.)
14. Assign variables (particularly variables which are to be used as constants, if that makes sense!) in immediate mode. This means you simply type in the required values for each variable using a normal LET statement, but do not use a line number. The ZX81 will respond with the pretty meaningless report code 0/0, but what it has done in fact is store the variable names and values, and you may refer to them in a subsequent program. The only limitations are that you must not press Run, Clear or New after assigning variables in this way (to start the program, use GOTO 0). You can, however, SAVE the program *and* the variables. (This means that even if pre-assigned variables do change their values during the course of a program, it is always possible to 're-run' by re-LOADing).
15. Finally, one last space-saving tip – use machine code. I know that this is about as useful to many users as saying "write your programs in Greek", but even if you cannot (yet!) write entire programs in machine code, why not start by using one or two small routines to replace lines of BASIC? For instance, here is a very common line which sets C equal to the CODE of the character at the top left of the screen:

```
10 LET C = PEEK (PEEK 16396
+ 256 * PEEK 16397 + 1)
```

And here is the same thing in machine code:

```
1 REM E£RND 7▚?TAN
10 LET C = USR 16514
```

In op-code form, this is:

```
2A 0C 40    LD HL,(16396)
23          INC HL
06 00       LD B,0
4E          LD C,(HL)
C9          RET
```

You can enter the REM statement directly from the keyboard provided that you remember:
a) RND and TAN are function words.
b) There is a space after the graphics character (which is the one on the T key), but not after RND.
c) POKE 16520,78 should be entered (in immediate mode) so that the '?' character will represent the correct op-code.

Just in case you are still thinking that all of this is purely academic, here is a program example which illustrates some of the above techniques and shows just how much you can fit into 1K.

Guillotine

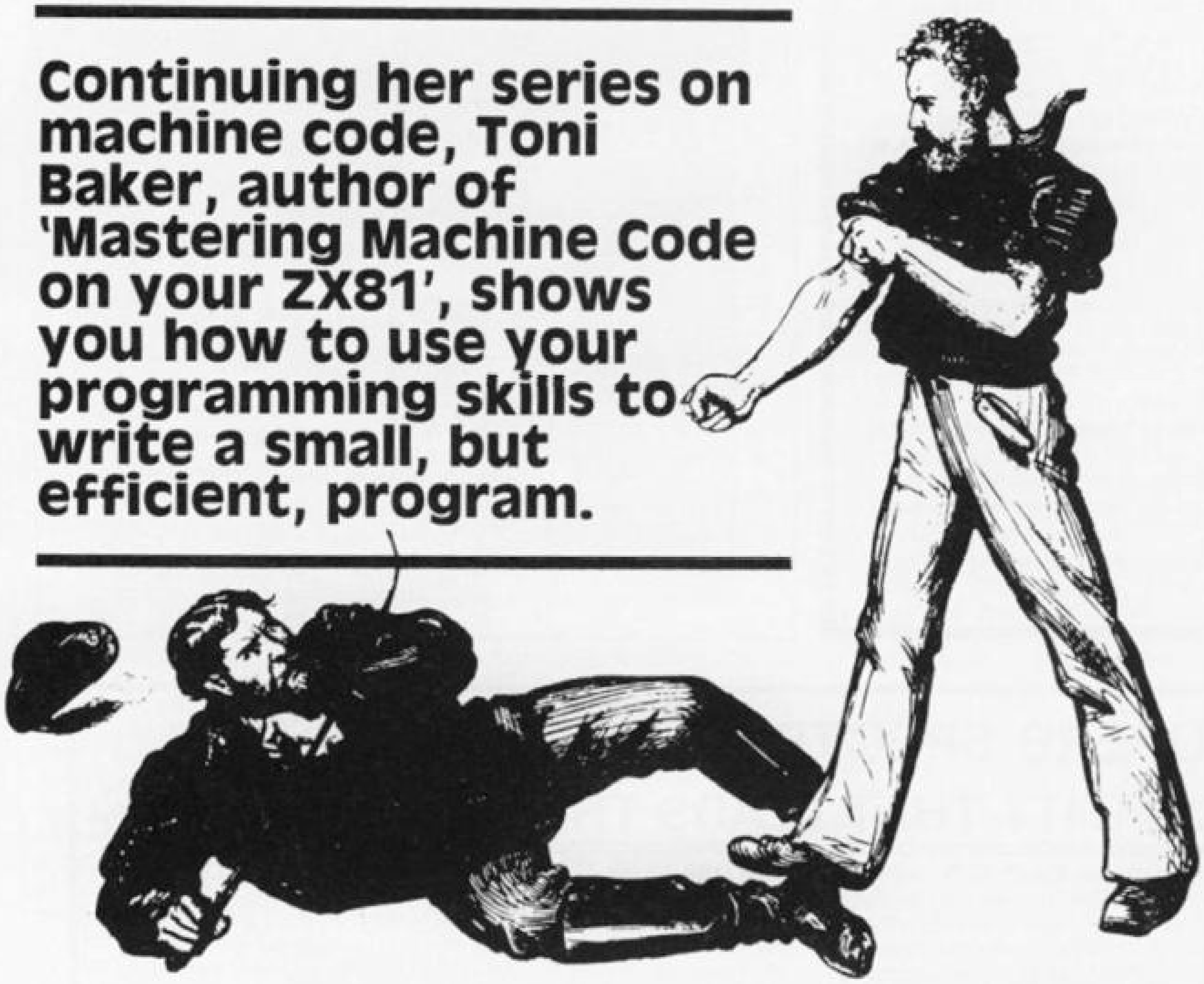
This program was written by Jean Hartopp. Jean is into computer graphics and 'Guillotine' makes good use of the ZX81 screen display to enhance the old 'standard' computer game where the ZX81 selects a number (in the range 1-100) which you have to guess in six attempts. The program saves on text by using the characters '<' and '>' to indicate whether your guess was too high or too low. This game will appeal to sadists who deliberately lose at Hangman just so that they can see the victim swing!

```
5 LET A=INT (RND*100)+1
10 FOR F=A/A TO 12
15 PRINT "#   #"
20 NEXT F
25 PRINT "##### 2121"
30 PRINT AT 8,7;"85"
35 PRINT TAB 6;"8HH5"
40 PRINT TAB 6;"7HG1"
45 PRINT "#-O-#  58"
50 FOR F=-5 TO 7
55 PRINT AT 10,5;"/"
60 PRINT AT F,A/A;"    "
65 PRINT "#---"
70 PRINT "#  /"
75 PRINT "# /"
80 PRINT "#/"
85 IF F>A-A THEN GOTO
    125
90 PRINT "# "
95 PRINT AT 10,5;"-"
100 INPUT G
105 IF A=G THEN GOTO 150
110 LET A$=">"
115 IF G>A THEN LET
    A$="<"
120 PRINT TAB 12;A$
125 NEXT F
130 PRINT AT 11,2;"6-"
135 FOR F=A/A TO 8
140 PRINT AT 13,F;" 0"
145 NEXT F
150 PRINT AT 11,12;A
155 INPUT A$
160 CLS
165 RUN
```

(Note that all the numbers and the letters, H and G, which appear in strings represent their respective graphic characters. The '#' symbol represents inverse space. All the other characters should be entered as listed.)

Mastering machine code on your Spectrum — part 3

Continuing her series on machine code, Toni Baker, author of 'Mastering Machine Code on your ZX81', shows you how to use your programming skills to write a small, but efficient, program.



In this article I shall introduce you to a few simple machine code instructions and use them to write a short but very, very impressive program.

The first two of these instructions are CALL and RET. RET you've seen before — it is used to end a machine code program in order to RETurn to BASIC, but it also has another use. CALL and RET in machine code are quite similar to GOSUB and RETURN in BASIC. CALL is used to enter a subroutine, and RET will exit the subroutine and move control back to the instruction immediately after the CALL instruction. Instead of a line number you have an address, so CALL pq means 'GOSUB address pq' — not 'GOSUB line pq'. In machine code, remember, there are no line numbers.

And now for some more exciting machine code instructions: IN and OUT. Both of these have equivalents in Spectrum BASIC (although not in ZX81 BASIC) but most people have absolutely no idea how to use them. In BASIC, they are in a sense similar to PEEK and POKE, and so in machine code there is a certain similarity between IN and OUT, and LD r,(pq) and LD (pq),r (r means any eight-bit register, and pq means any address). Let's first look at the formats of these statements as shown below. Notice that the standard op-codes are incomplete. For instance, there is an instruction called IN D,(C); however, this is really just a shorthand notation — what it *really* means is IN D,(BC). Also note that in the above table, A is a register and n is a fixed constant. Now let's get down to what these statements actually *do*.

Machine Code	Short For	Basic Equivalent
IN A,(n)	IN A,(An)	LET A=IN (256 ✱ A+n)
IN r,(C)	IN r,(BC)	LET r=IN (256 ✱ B+C)
OUT (n),A	OUT (An),A	OUT 256 ✱ A+n,A
OUT (C),r	OUT (BC),r	OUT 256 ✱ B+C,r

What's it all about?

IN fetches a number from the outside world and loads this number into a machine code register. OUT takes a number from a machine code register and sends it to the outside world. There are a number of ways in which the Z80 chip can communicate with the outside world. One way is via the keyboard. It can receive information from the keyboard using IN instructions as shown in the table on the page opposite. The number you actually get from such an IN instruction is best viewed in binary. Each such IN instruction tests five specific keys on the keyboard, and for each key produces a 1 if the key is *not* pressed and a O if the key *is* pressed. Bits 7, 6 and 5 will contain rubbish, bits 4, 3, 2, 1 and O test the keys in the order given above. For instance, suppose keys Symbol Shift and B were held down simultaneously, then IN r,(7FFE) would produce the (binary) number xxx01101. The zeroes indicate that a key is pressed. The xs are rubbish and their value is unimportant. You can achieve this test in two ways: either LD BC,7FFE/IN r,(C) or LD A,7F/IN A,(FE). See if you can deduce why both of these are equivalent to "IN r,(7FFE)".

Another way of communicating with the outside world is via the television screen — this is an *output device* (the keyboard is an *input device*). The contents of the screen itself are controlled by addresses in memory (4000 to 5AFF), but the colour of the BORDER is actually changed by an OUT instruction. Try typing (in BASIC) OUT 254,6 and see what happens. Now press Enter on its own. What's happening is that the ROM itself is re-setting the border colour (with its own OUT instruction) to whatever number is given by the system variable BORDCR. Try typing POKE 23624,6 ✱ 8 and see what happens. Now press Enter on its own.

The instruction, OUT (xxFE),r, in any form (where the xx can be any byte whatsoever — its value is irrelevant), will instantly change the border colour to the colour whose code is contained in the register given. For example, LD A,02/LD C,FE/OUT (C),A will change the border colour to red. Notice that we do not need to worry about the B register because in this particular case its value is unimportant. This is a side-effect of the Spectrum hardware, not of machine code itself.

IN r,(FEFE) scans section 0:	V, C, X, Z and Caps Shift
IN r,(FDFE) scans section 1:	G, F, D, S and A
IN r,(FBFE) scans section 2:	T, R, E, W and Q
IN r,(F7FE) scans section 3:	5, 4, 3, 2 and 1
IN r,(EFFE) scans section 4:	6, 7, 8, 9 and 0
IN r,(DFFE) *scans* section 5:	Y, U, I, O and P
IN r,(BFFE) scans section 6:	H, J, K, L and Enter
In r,(7FFE) scans section 7:	B, N, M, Symbol Shift and Space

All change

To change tact completely I would like to introduce you to a new set of instructions altogether — the ROTATE and SHIFT instructions. In order to understand the effect of these instructions it is useful to think of the numbers involved in binary rather than Hex (useful, although not necessary). Suppose A contains the number 00101101, and the carry flag (which I shall refer to as K) contains the number 1. The instruction RLA (which stands for Rotate Left A) will rotate every bit one position to the left. The leftmost bit of A moves into K, which itself moves into the rightmost position of A. In the example given, the result would be that A contained the number 01011011 and K would now be zero. On the other hand, if RRA were used instead of RLA (guess what RRA stands for!) then A would contain 10010110 and K would contain 1. You can do the same for any register except that the instructions are written differently — you write 'RL B' (with a space) instead of 'RLB' (without a space). In fact, there is actually an instruction called 'RL A' (with a space) which at first glance appears to do exactly the same as 'RLA' (without a space). The difference is that RL A will change the value of the Z flag, whereas RLA will not.

Flags are funny things. Their concept is quite simple — a flag is a one-bit register, or a register which can only store the numbers zero and one. There are four flags we can make use of: the carry flag, K, we've already seen; the zero flag, Z — if an instruction changes it (some don't), it will become 1 if the result of the instruction is zero, and 0 if the result of the instruction is not zero; the sign flag, S, becomes 0 for a positive answer, and 1 for a negative answer (in other words it actually equals bit 7 of the answer); and the parity/overflow flag, P, is changed in two ways: for arithmetic instructions it becomes 1 if the number increases from 7F or less to 80 or more, or if the number decreases from 80 or more to 7F or less — for non-arithmetic instructions it is set to 1 if the result contains an even number of ones, or zero if it contains an odd number of ones.

What is the point of the flag P? Its non-arithmetic use (parity check) is absolutely useless as far as I can see. Its arithmetic use (overflow check) comes into play when you're doing sums. It assumes that numbers 80 to FF are all negative (FF being minus one, FE being minus two, and so on), and so since 42 (positive) plus 42 (positive) equals 84 (negative?) — we have an overflow!

Back to the ROTATE instructions. As well as RL and RR, there is another breed of rotate instructions — RLC and RRC, which stand for Rotate Left (or Right) without Carry. If A started off as 00101101, then irrespective of the original value of the carry, RLCA would change it to 01011010 and RRCA would change it to 10010110. K *would*, in fact, be changed — its new value would be the bit which came off one end and emerged at the other, but its original value does not come into play. As before, you should note that RLCA is subtly different to RLC A, and that RLCB (etc) do not exist wheras RLC B (etc) do.

Best of three

There are three different types of SHIFT instruction, and *all* of them are written with a space. The first is called SLA — the A stands for 'Arithmetic', so you have to put the name of a register as well, eg SLA A or SLA B. SLA is a little bit like RL except that instead of the carry moving into the rightmost bit of the register, it just disappears altogether. The rightmost bit of the register is instead always reset to zero. In effect, the value of the register has been multiplied by two. There are two similar instructions to shift right which divide the appropriate register by two. We have to remember though that numbers between 80 and FF can each have two different meanings; for instance, FE can mean either 254 or minus two. Now, 254 divided by two is 127 (in Hex 7F) whereas minus two divided by two is minus one (in Hex FF) — so there must be a simple way to overcome this problem. There are two different Shift Right instructions, called SRA and SRL. SRA (Shift Right Arithmetic) treats numbers between 80 and FF as negative and divides them by two. Its precise effect is to move each bit one position to the right, with the rightmost bit moving into the carry, and the leftmost bit remaining unchanged (bit 6 and bit 7 are always identical immediately after a SRA instruction). SRL (Shift Right Logical) treats numbers between 80 and FF as positive and divides them by two. Its precise effect is to move each bit one position to the right, with the rightmost bit moving into the carry, and the leftmost bit becoming zero.

There is one more instruction I need to give you before the program will make sense — the instruction HALT. HALT in machine code is quite similar to PAUSE 1 in BASIC. Its precise effect is to wait (for a maximum of 1/50th of a second) until the next TV frame has been output to the screen, and then to continue from the next instruction. Its use, as I have made use of it, is in synchronisation — avoiding that horrid jump you sometimes get when you change border colours halfway through the screen outputting.

The program I promised you is called Tricolours and is listed separately. You now have sufficient knowledge to be able to understand the listing. I suggest that you read the listing and see if you can work out exactly what it does, and *then* type it in to see if you were right. I think you will find it reasonably impressive, and, in fact, a rather surprising display coming as it does from a ZX Spectrum.

Programming techniques

There are several ways to learn machine code. One way is for me to give you a program and explain why it works. Another way is for me to write a program and *not* explain how it works so that you have to work it out for yourselves — unfortunately, deciphering other people's programs (especially mine) turns out to be quite tricky. Several people have commented that they've learned far more machine code from ironing out the bugs in my programs than they have from the ones which worked, and actually wondered if the bugs were put there deliberately! (I confess they weren't — this brilliant teaching method occurs purely as a consequence of typing errors and the like!).

However, there is one last technique by which to learn machine code, and that is to write it yourself. You take a scrap of paper and scribble down a few ideas, turn the ideas into some sort of coherent pattern of how you think it ought to work (either with or without a flow diagram — it doesn't matter), elaborate it into a program or part of a program, turn it into Hex, feed it in, watch it crash, turn around three times, jump up and down, and sing the National Anthem wondering why it went wrong!

Crash landing

Programs go wrong all the time as you're writing them, especially in machine code. The difference is that in machine code a crash is usually fairly fatal, sometimes even meaning that you have to actually switch the machine off before it will behave sensibly. (The Spectrum usually behaves quite sensibly when it's switched off.) All I'm really trying to say is that when this happens (note: *when*, not *if*) that you shouldn't be too disheartened and that you should just go back and try again, and try to find out what the error is. Usually there *is* an error — it is quite unusual for the breakdown to be caused by a glitch in the Spectrum itself, especially if you happen to find a particular glitch which only ever occurs when you try to play with machine code.

The best possible advice I can give as regards writing programs is to make all of your programs as short as you possibly can. This will train you to write programs efficiently. Making programs short involves putting all of the machine code instructions to their best possible use, and sometimes using clever little tricks to save space; for instance, using SBC HL,HL, instead of LD HL,0000 when you know that the carry is zero. One really useful instruction which really does save space is the instruction LDIR (LD, Increment and Repeat) which is equivalent to a whole set of instructions: "LD(DE),(HL)"/INC ▶

Tricolours

Use HEXLD3 to load this into the ZX Spectrum. The BASIC part of the listing is as follows:

```
700  BORDER 1: CLS
710  FOR i=1 TO 6: PRINT PAPER 2,TAB 0: NEXT i
720  FOR i=1 TO 12: PRINT PAPER 7,TAB 0: NEXT i
730  FOR i=1 TO 4: PRINT PAPER 1,TAB 0: NEXT i
740  RANDOMIZE USR 32768
750  STOP
```

And here is the machine code part of the listing (to address 8000).

```
01FE7F   START    LD BC,7FFE
76       LOOP     HALT          Wait for next TV frame.
3E02              LD A,yellow
219D03            LD HL,039D
CD1E80            CALL STRIPE
3E07              LD A,white
213A03            LD HL,033A
CD1E80            CALL STRIPE
3E01              LD A,blue
ED79              OUT (C),A
ED78              IN A,(C)      Remember, it's really IN A,(BC).
1F                RRA           K= test on Space key.
38E6              JR C,LOOP     Loop if Space key not pressed.
C9                RET           Return to BASIC.
ED79     STRIPE   OUT (C),A     This is a subroutine.
2B       DELAY    DEC HL        HL is the timing for a delay loop.
7C                LD A,H
B5                OR L          Is HL=0?
20FB              JR NZ,DELAY   Keep looping until HL=0.
C9                RET           End of subroutine.
```

Use RUN 700 to RUN this program. Press the Space key to exit from it.

Newcols

```
78       NEWCOLS  LD A,B        A = PAPER colour.
76                HALT          Synchronise with TV frame.
D3FE              OUT (FE),A    Change BORDER colour to
                                new PAPER colour.
78                LD A,B        A=paper colour.
87                ADD A,A
87                ADD A,A
87                ADD A,A       Multiply A by eight.
81                ADD A,C       Add INK colour.
210058            LD HL,5800    Point HL to first attribute byte.
110158            LD DE,5801    Point DE to second attribute byte.
01FF02            LD BC,02FF    BC = number of attribute bytes,
                                excluding the first one.
77                LD (HL),A     Set first attribute byte.
EDB0              LDIR          Set all remaining attribute bytes.
C9                RET           End of subroutine.
```

To run from BASIC, add these machine code instructions, POKE into the first statement, and CALL from the label START using USR:

```
010000   START    LD BC,????    Set PAPER and INK colours.
18E5              JR NEWCOLS    Execute machine code above.
```

For use with FN U:

```
2A0B5C   PARAMS   LD HL,(DEFADD)
110400            LD DE,0004
19                ADD HL,DE     Point HL to PAPER colour.
4E                LD C,(HL)     C = PAPER colour.
19                ADD HL,DE
19                ADD HL,DE     Point HL to INK colour.
46                LD B,(HL)     B = INK colour.
18D8              JR NEWCOLS    Execute machine code above.
```

HL/INC DE/DEC BC/repeat until BC=0, and can be used to copy several bytes at once from one address to another. Study the program Newcols which uses this instruction — it instantly changes the foreground and background colours on the screen to those colours contained in registers B and C.

Notice that in order to use Newcols as it stands now, you have to do two POKE instructions before RUNning it in order to define what the foreground and background colours will be. Although this is not difficult, I personally consider it a nuisance, and a far, far better programming technique is as follows.

Insert the BASIC line DEF FN U(X,Y)=USR address of label PARAMS at any point in the program. Now you have complete control — you may change the PAPER and INK colours at will using statements like LET L=FN U(2,6) or LET L=FN U(7,INT (4 * RND)). The reason this works is because a DEF FN statement line is actually POKEd for us by the ROM. When a reference to FN is made, the values for it are calculated by the ROM, POKEd into the DEF FN statement, and then calculated. Once POKEd, the BASIC DEF FN statement looks like this (those bytes underlined are invisible from the BASIC listing):

DEF FN U(X OE xx xx xx xx xx, Y OE yy yy yy yy yy) = function expression.

The bytes I have written as xx xx xx xx xx and yy yy yy yy yy are the five byte Sinclair form of the two numbers in the FN U expression. If the two numbers are both integers between 00 and FF (which in this case they will be) then the Sinclair forms will actually be 00 00 xx 00 00 and 00 00 yy 00 00. The system variable DEFADD points to the first variable name inside the brackets (in this case to the x) so that LD HL,(DEFADD)/INC HL/INC HL will point HL to the first byte of the Sinclair form of the first number.

You can use this technique not just for this program, but for *any* machine code program which requires the passing of parameters from BASIC.

With that I shall leave you for now. In my next article, I'll be putting some concentration into efficient use of the stack, among other things.

Mastering machine code on your Spectrum — part 2 continued . . .

In the last issue of ZX Computing, we were unable to reproduce two of the tables Toni promised you in the article. The space we had allocated for the tables proved insufficient so, rather than squeeze them in in incredibly small type, we decided to publish half last month and the rest below. We apologise for any inconvenience this may have caused, but we hope you'll agree it was worth waiting for!

TABLE THREE

c	NZ	Z	NC	C	PO	PE	P	M
CALL c,pq	C4qqpp	CCqqpp	D4qqpp	DCqqpp	E4qqpp	ECqqpp	F4qqpp	FCqqpp
JP c,pq	C2qqpp	CAqqpp	D2qqpp	DAqqpp	E2qqpp	ECqqpp	F2qqpp	FAqqpp
JRc,e	20ee	28ee	30ee	38ee	—	—	—	—
RET c	CO	C8	DO	D8	EO	E8	FO	F8

Machine Code Instructions

INSTRUCTIONS Op-code	Hex-code	FLAGS S	Z	–	H	–	P	N	C
ADC A,r	table 1	@	@	–	@	–	@	0	@
ADC HL,s	table 2	@	@	–	@	–	@	0	@
ADD A,r	table 1	@	@	–	@	–	@	0	@
ADD HL,s	table 2	–	–	–	@	–	–	0	@
ADD IX,s	table 2	–	–	–	@	–	–	0	@
ADD IY,s	table 2	–	–	–	@	–	–	0	@
AND r	table 1	@	@	–	1	–	@	0	0
BIT b,r	table 1	?	@	–	1	–	@	0	0
CALL pq	CDqqpp	–	–	–	–	–	–	–	–
CALL c,pq	table 3	–	–	–	–	–	–	–	–
CCF	3F	–	–	–	x	–	–	0	@
(the H flag becomes the previous value of the C flag)									
CP r	table 1	@	@	–	@	–	@	1	@
CPI	EDA1	@	x	–	@	–	x	1	–
CPD	EDA9	@	x	–	@	–	x	1	–
CPIR	EDB1	@	x	–	@	–	x	1	–
CPDR	EDB9	@	x	–	@	–	x	1	–
(Z becomes 1 if BC becomes zero, P/V becomes 1 if A = (HL-1))									
CPL	2F	–	–	–	1	–	–	1	–
DAA	27	@	@	–	@	–	@	–	@
DEC r	table 1	@	@	–	@	–	@	1	–
DEC s	table 2	–	–	–	–	–	–	–	–
DI	F3	–	–	–	–	–	–	–	–
DINZ e	10ee	–	–	–	–	–	–	–	–
EI	FB	–	–	–	–	–	–	–	–
EX AF, AF'	08	–	–	–	–	–	–	–	–
EX DE,HL	EB	–	–	–	–	–	–	–	–
EX (SP),HL	E3	–	–	–	–	–	–	–	–
EX (SP),IX	DDE3	–	–	–	–	–	–	–	–
EX (SP),IY	FDE3	–	–	–	–	–	–	–	–
EXX	D9	–	–	–	–	–	–	–	–
HALT	76	–	–	–	–	–	–	–	–
IM 0	ED46	–	–	–	–	–	–	–	–
IM 1	ED56	–	–	–	–	–	–	–	–
IM 2	ED5E	–	–	–	–	–	–	–	–
INC r	table 1	@	@	–	@	–	@	0	–
INC s	table 2	–	–	–	–	–	–	–	–
IN A,(n)	DBnn	–	–	–	–	–	–	–	–
IN r,(C)	table 1	@	@	–	@	–	@	0	–
INI	EDA2	?	x	–	?	–	?	1	–
IND	EDAA	?	x	–	?	–	?	1	–
(Z becomes 1 if B becomes zero)									
INIR	EDB2	?	1	–	?	–	?	1	–
INDR	EDBA	?	1	–	?	–	?	1	–
JP pq	C3qqpp	–	–	–	–	–	–	–	–
JP c,pq	table 3	–	–	–	–	–	–	–	–
JP (HL)	E9	–	–	–	–	–	–	–	–
JP (IX)	DDE9	–	–	–	–	–	–	–	–
JP (IY)	FDE9	–	–	–	–	–	–	–	–
JR e	18ee	–	–	–	–	–	–	–	–
JR c,e	table 3	–	–	–	–	–	–	–	–
LD(BC),A	02	–	–	–	–	–	–	–	–
LD A,(BC)	0A	–	–	–	–	–	–	–	–
LD (DE),A	12	–	–	–	–	–	–	–	–
LD A,(DE)	1A	–	–	–	–	–	–	–	–
LD I,A	ED47	–	–	–	–	–	–	–	–
LD R,A	ED4F	–	–	–	–	–	–	–	–
LD A,I	ED57	@	@	–	0	–	x	0	–
LD A,R	ED5F	@	@	–	0	x	0	–	
(P/V is set to interrupt storage flag)									
LD SP,HL	F9	–	–	–	–	–	–	–	–
LD SP,IX	DDF9	–	–	–	–	–	–	–	–
LD SP,IY	FDF9	–	–	–	–	–	–	–	–

INSTRUCTIONS Op-code	Hex-code	FLAGS S	Z	–	H	–	P	N	C
LD r,r	table 1	–	–	–	–	–	–	–	–
LD s,mn	table 2	–	–	–	–	–	–	–	–
LD A,(pq)	3Aqqpp	–	–	–	–	–	–	–	–
LD s,(pq)	table 2	–	–	–	–	–	–	–	–
LD (pq),A	32qqpp	–	–	–	–	–	–	–	–
LD (pq),s	table 2	–	–	–	–	–	–	–	–
LDI	EDA0	–	–	–	0	–	x	0	–
LDD	EDA8	–	–	–	0	–	x	0	–
(P/V becomes 0 if BC becomes 0)									
LDIR	EDB0	–	–	–	0	–	0	0	–
LDDR	EDB8	–	–	–	0	–	0	0	–
NEG	ED44	@	@	–	@	–	@	1	@
NOP	00	–	–	–	–	–	–	–	–
OR r	table 1	@	@	–	0	–	@	0	0
OUT (n),A	D3nn	–	–	–	–	–	–	–	–
OUT (C),r	table 1	–	–	–	–	–	–	–	–
OUTI	EDA3	?	x	–	?	–	?	1	–
OUTD	EDAB	?	x	–	?	–	?	1	–
(Z becomes 1 if B becomes zero)									
OTIR	EDB3	?	1	–	?	–	?	1	–
OTDR	EDBB	?	1	–	?	–	?	1	–
POP AF	F1	x	x	x	x	x	x	x	x
(Flags are determined by the byte at the top of the stack)									
POP s	table 2	–	–	–	–	–	–	–	–
PUSH AF	F5	–	–	–	–	–	–	–	–
PUSH s	table 2	–	–	–	–	–	–	–	–
RES b,r	table	–	–	–	–	–	–	–	–
RET	C9	–	–	–	–	–	–	–	–
RET c	table 3	–	–	–	–	–	–	–	–
RETN	ED45	–	–	–	–	–	–	–	–
RETI	ED4D	–	–	–	–	–	–	–	–
RLA	17	–	–	–	0	–	–	0	@
RL r	table 1	@	@	–	0	–	@	0	@
RLCA	07	–	–	–	–	–	–	–	–
RES b,r	table 1	–	–	–	–	–	–	–	–
RET	C9	–	–	–	–	–	–	–	–
RET c	table 3	–	–	–	–	–	–	–	–
RETN	ED45	–	–	–	–	–	–	–	–
RETI	ED4D	–	–	–	–	–	–	–	–
RLCA	07	–	–	–	0	–	–	0	@
RRCA	0F	–	–	–	0	–	–	0	@
RLA	17	–	–	–	0	–	–	0	@
RRA	1F	–	–	–	0	–	–	0	@
RLC r	table 1	@	@	–	0	–	@	0	@
RRC r	table 1	@	@	–	0	–	@	0	@
RL r	table 1	@	@	–	0	–	@	0	@
RR r	table 1	@	@	–	0	–	@	0	@
RRD	ED67	@	@	–	0	–	@	0	–
RLD	ED6F	@	@	–	0	–	@	0	–
RST 00	C7	–	–	–	–	–	–	–	–
RST 08	CF	–	–	–	–	–	–	–	–
RST 10	D7	–	–	–	–	–	–	–	–
RST 18	DF	–	–	–	–	–	–	–	–
RST 20	E7	–	–	–	–	–	–	–	–
RST 28	EF	–	–	–	–	–	–	–	–
RST 30	F7	–	–	–	–	–	–	–	–
RST 38	FF	–	–	–	–	–	–	–	–
SBC A,r	table 1	@	@	–	@	–	@	1	@
SBC HL,s	table 2	@	@	–	@	–	@	1	@
SCF	37	–	–	–	0	–	–	0	1
SET b,r	table 1	–	–	–	–	–	–	–	–
SLA r	table 1	@	@	–	0	–	@	0	@
SRA r	table 1	@	@	–	0	–	@	0	@
SLR r	table 1	@	@	–	0	–	@	0	@
SUB r	table 1	@	@	–	@	–	@	1	@
XOR r	table 1	@	@	–	0	–	@	0	0

Competition

Win some software for your Spectrum in this easy-to-enter competition

Here's a chance for you to win some software for your ZX Spectrum.

And it's easy! All you have to do is to make as many words of four letters and over out of the Spectrum keyword 'RANDOMIZE'. All the words must be in the English language and will be checked against those listed in The Concise Oxford Dictionary of Current English, sixth edition.

Fill in the words you have found in the space provided on the coupon, along with your name and address, and post off your entry to us. But before you do, make sure you write the number of words you found, on the back of the envelope. The coupon provided has room for 40 words — should you find more words than you can fit on the coupon, you may attach a separate sheet (but you must still enclose the original coupon or a photocopy).

There will be three winners to this competition — either the three entries with the highest number of words or in the case of more than three entries sharing the highest number of words, these entries will be placed in a hat and the winners will be the first three chosen at random.

The prizes

The three winners will each receive one of the following three cassettes by Artic Computing.

Spectrum Chess (48K) There are six levels of difficulty in this version of chess. Including the facility to 'castle' and 'en passent', the program also allows you to alter the level of play or even swap colours during a game!

Spectrum Bug (16K/48K) This program is a machine code monitor and debugger. Occupying 1½K of memory, the program provides you with 16 new commands for your Spectrum.

Planet of Death (16K/48K) In this adventure game, you find yourself stranded on an alien planet separated from your spaceship. Your mission is to overcome the hazards and dangers of the planet, and find your spaceship. This cassette is also one of the new range of software released by Sinclair Research.

Rules

This competition is open to all UK and Northern Ireland readers of ZX Computing except employees of Argus Specialist Publications Ltd, their printers and distributors, employees of Sinclair Research Ltd and their distributors, employees of Artic Computing and their distributors, or anyone else associated with the competition.

As long as the correct coupon is used for each entry, there is no limit to the number of entries.

All entries must have the number of words found written on the outer flap of the envelope. Entries without this number will not be accepted.

All entries must be postmarked before the 31st March, 1983.

The prizes will be awarded to the three entries with the highest number of correct words found, or in the case of a tie, three of the highest entries will be chosen at random.

All words found must be within The Concise Oxford Dictionary of Current English, sixth edition.

No correspondence will be entered into with regard to the results and it is a condition of entry that the Deputy Editor's decision is accepted as final.

The winners will be notified by post and the results will be published in a future issue of ZX Computing.

Address your answers to:

ZX Computing Competition, 145 Charing Cross Road, London WC2H 0EE.



Name . . . .

Address . . . .

. . . .

. . . . P



Derby day



Have a day at the races with Terry Mills of Stockport.

In this program for the unexpanded ZX81, there are four horses (denoted by inverse numbers) running in a race.

Here is a breakdown of the listing to show you how the program works:

Line 5 Displays the start and finishing posts of the course.
Lines 10-50 Set up the variables.
Line 55 Prints a space over the old position of the horses.
Lines 60-80 Calculate the random movement of the horses.
Line 81-82 Introduce a slight delay before the next movement of the horses is made.
Lines 85-90 Print the horses in their new position in the race.
Line 95 Decides if a horse has won the race; if a particular horse has won, the winner is displayed and the game ends.
Line 100 If none of the horses have reached the finishing post, the program returns to print the new position of the horses.

Should you wish to speed up the race, lines 81 and 82 may be omitted. Should you have more than 1K available, you may wish to include a short routine to allow players to have small bets on the different horses.

```
5   PRINT AT 0,0;"-START
    -";AT 20,0;"-FINISH-"
10  LET A=1
15  LET B=0
20  LET C=1
25  LET D=2
30  LET E=1
35  LET F=4
40  LET G=1
45  LET H=6
50  LET Z=19
55  PRINT AT A,B;" ";AT
    C,D;" ";AT E,F;" ";AT
    G,H;" "
60  LET X=INT(RND*5)
65  IF X=1 THEN LET A=A
    +1
70  IF X=2 THEN LET C=C
    +1
75  IF X=3 THEN LET E=E
    +1
80  IF X=4 THEN LET G=G
    +1
81  FOR N=1 TO 10
82  NEXT N
85  PRINT AT A,B;"1";AT
    C,D;"2"
90  PRINT AT E,F;"3";AT
    G,H;"4"
95  IF A=Z OR C=Z OR
    E=Z OR G=Z THEN
    PRINT AT 11,15;
    "WINNER IS ";Z;Q
100 GOTO 55
```

Bookshelf

With so many publications being written for the Spectrum, which one should you look at first? Our review panel take a brief look at ten of the new titles to help you make your choice.

Spectrum Machine Language For The Absolute Beginner — Edited by William Tang

This book is designed as an introductory text to the field of machine and assembly language programming for the ZX Spectrum.

Inside the book, there are five section headings, each of which has been split up into a number of sub-sections. The first section is called 'Finding your way around in machine language' and starts from the very beginning, assuming absolutely no prior knowledge of working with machine language. From the basics, though, you quickly move onto how to manipulate the stack, loops and jumps, use of subroutines and block operations.

The second section of the text deals with 'Instructions that are less frequently used' and covers register exchanges, rotates and shifts, interrupts and restarts, and many others. As in the first section of the book, the text is liberally sprinkled with examples for you to try and most of the sections end with either a summary of the information in that section or a number of exercises for readers to work their way through.

The next section, 'Programming your Spectrum', deals first with the planning of a program; this is fairly comprehensive, dealing mainly with the 'top-down' approach to disciplining your programming. Then the authors take a look at the Spectrum itself, paying particular attention to the keyboard, the video screen display and the sound output.

The last two sections concentrate on actual programs listed within the text. Looking first at monitor programs, there are two provided: EZ-Code Machine Language Editor and HexLoad Machine Code Monitor. As well as clearly reproduced listings, the programs are accompanied by an explanation of each.

The last section concentrates on the production of one program, Freeway Frog, which is listed over 36 pages of the book. From first programming principles, the structure of the program is planned, developed and finally put together to form a complete listing.

There are seven appendices, with useful data you'll need when you begin experimenting with machine and assembly language yourself.

Spectrum Machine Language For The Absolute Beginner, edited by William Tang, is published by Melbourne House. The book has 243 pages and is priced at £6.95. ISBN 0 86161 110 1



Games ZX Computers Play — Edited by Tim Hartnell

Recently crowned the 'Barbara Cartland of the computer book field', Tim Hartnell here produces a book for the Spectrum and ZX81 which would fall into the 'fun' section of his range of books.

Containing 15 programs for the ZX Spectrum and 15 for the ZX81, all the listings are direct from the ZX Printer, well reproduced and guaranteed to run. The emphasis is on moving graphics, intelligent play by the computer, user-defined graphics where needed, and providing the reader with programming ideas which can either be adapted, converted or improved on.

The book contains a variety of programs contributed by a number of experienced programmers. There is a Spectrum version of a three-dimensional maze, and many arcade-type games such as Breakout, Zombies and Quack Attack. There are also the wide range of card games and board games.

Each program is accompanied with a brief explanation of how the program works. In some cases, the programs are illustrated with screen display dumps of the game being played.

Games ZX Computers Play, edited by Tim Hartnell, is published by Interface Publications. The book has 169 pages and is priced at £3.25. ISBN 0 907563 13 9

The Spectrum Pocket Book — Trevor Toms

Continuing the 'Pocket Book' series, this book goes further than its two predecessors about the ZX80 and ZX81, providing the reader with programs which are fun, serious, educational, and a number of 'useful' tools for your Spectrum.

The programs in the first part of the book, the BASIC section, cover a variety of subjects. Ranging from the obvious games programs such as Robot Chase, 3D Maze and Reversi, there are also a number of 'serious' listings such as Budget Account, Debugging Programs, Useful Subroutines and User Graphics Tablet.

The second part of the book, the Machine Code section, begins with an introduction to machine code followed by a comprehensive section relating the use of machine code directly to your ZX Spectrum. Also included in this section are ZXASM, a Symbolic Assembler; ZXDISASM, a Symbolic Disassembler; and ZXMCMON, a machine code monitor. An ap-

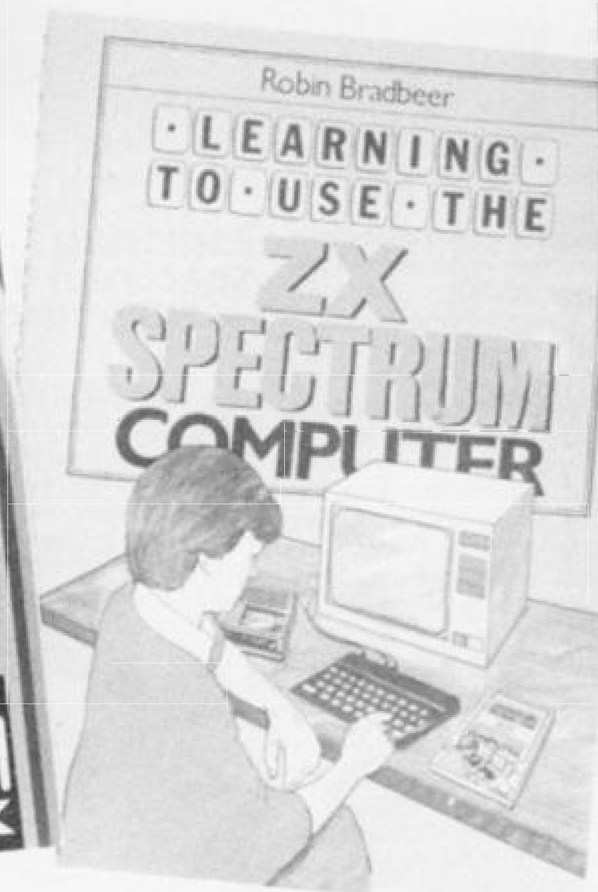
pendix at the back of the book contains a screen toolkit.

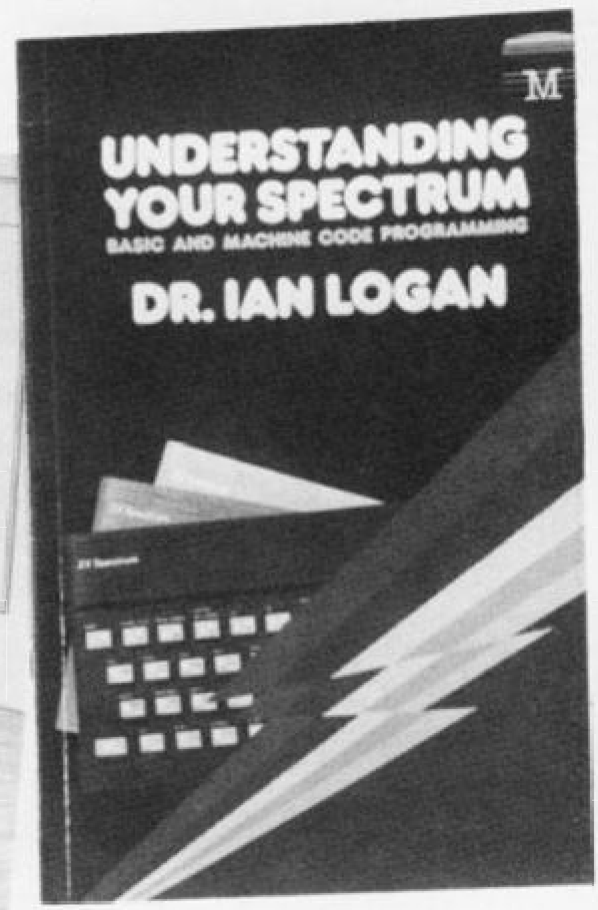
All the program listings have been produced using an RS232C interface which was specially designed (and available in design form from the author). The listings are fully annotated to help you understand the structure and thinking behind the program, and all the Spectrum keywords are printed in a bold typeface.

An explanation of each program is provided and the listings are clearly reproduced. All the programs are for the 16K ZX Spectrum.

The Spectrum Pocket Book, written by Trevor Toms, is published by Phipps Associates. The book has 160 pages and is priced at £6.50.
ISBN 0 950 7302 8 9.







The Working Spectrum — David Lawrence

This book is based on a collection of programs written on such areas as data storage, finance, calculation, graphics, household management and education.

Split into six chapters, the first chapter highlights the Spectrum as a filing cabinet. Providing a program called Unifile, any file containing records with a regular structure can be handled. The second chapter continues the theme of tying the Spectrum in with business, with three programs which are fairly self-explanatory: Budget, Accountant and Banker.

Chapter three concerns itself with Spectrum graphics with five programs which amongst other things help you create new characters, draw pictures and define a design of up to 65536 by 65536 pixels across. The fourth chapter sets the Spectrum up as a home tutor complete with three programs providing a multiple choice test with up to 1,000 different questions and answers, a multiple picture reading tutor, and a listing which tests your knowledge of geography.

The author has provided a collection of miscellaneous routines in chapter five, including Calculator, Calories, Graph, Renumber and Typist. Traditionally, all computer books should contain at least one game, and this book is no exception. Three games listings are included — Tracker, Missile and Word Sort.

Each listing is given as a series of modules. The modules are accompanied by an explanation of its function, a commentary on the individual lines, and suggestions are given to perform simple tests on the module to assure yourself that the module is correct before continuing to type any more in.

The Working Spectrum, written by David Lawrence, is published by Sunshine Books Ltd. The book has 216 pages and is priced at £5.95.
ISBN 0 946408 00 9

Learning To Use The ZX Spectrum Computer — Robin Bradbeer

This text is part of a general series of 'Learning To Use...' books which puts the emphasis on the beginner using the computer in work or leisure, rather than becoming a computer theorist.

Beginning the book is a chapter introducing the Spectrum, followed by a section on using the ZX Spectrum. This last chapter tells you in very simple and easy-to-follow instructions, how to begin your computing career on the machine. At the end of the chapter some self-test questions are included for the reader to assimilate how much experience has been gained.

Chapter three introduces the reader to some elementary programming skills, with details on how to SAVE programs on cassette and how to use the ZX Printer. Again, self-test questions are provided to help readers gauge their success.

The fourth chapter tackles the subject of graphics, concentrating on the screen and memory, patterns and the use of colour, movement and animation, user definable graphics and sound. Examples are included allowing the reader to illustrate the techniques with some practical programming.

Chapter five includes some of the special features of the ZX Spectrum with particular reference to the internal clock, special locations and how to examine them, and the user port. Three appendices are provided covering further reading, the differences between ZX81 and ZX Spectrum BASIC, and a glossary.

Learning To Use The ZX Spectrum Computer, written by Robin Bradbeer, is published by Gower Publishing Company Ltd. The book has 76 pages and is priced at £4.95.
ISBN 0 566 03481 6

Understanding Your Spectrum — Dr. Ian Logan

Dr. Logan claims this book has three main aims: to explain, in simple terms, how the Spectrum works; to teach Z80 code ▶

from first principles; and to give details of monitor entry points so that efficient programs may be written.

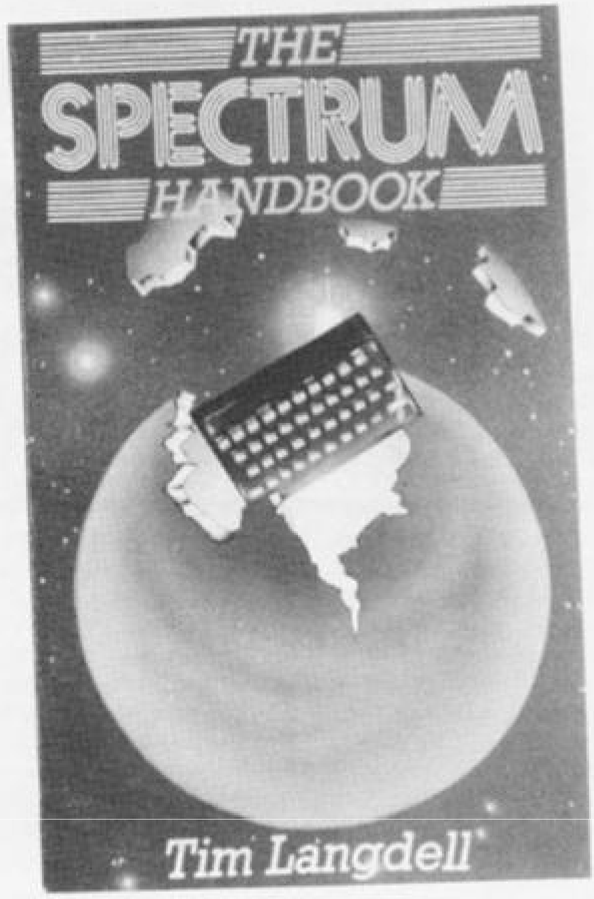
And the author does his best to explain all of it — no easy task! The first two chapters introduce commands and functions which Sinclair BASIC has to offer. Programming examples are provided throughout to aid readers in their understanding.

The third chapter moves into the realms of machine code providing an introduction to the Z80 microprocessor. Illustrations are used where possible to help with this explanation. This is followed by a section on the mathematics of machine code programming and one on the Z80 machine code instruction set.

It is at this stage that you are provided with some sample programs to use and understand. Twenty-two programs are provided in total. The final two chapters of the book concentrate on an outline of the 16K monitor program resident inside the Spectrum and how to utilise the monitor program's subroutines.

Four appendices are included at the back of the book complete with useful reference tables for Z80 machine code instructions, and decimal to hexadecimal conversion. There is also a list of the currently available machine code handling programs. The last appendix is a list of 11 programming errors in the 16K monitor program in the Spectrum, at least two of which are very useful to know.

Understanding Your Spectrum, written by Dr. Ian Logan, is published by Melbourne House. The book has 190 pages and is priced at £7.95.
ISBN 0 86759 114 5



The Spectrum Handbook — Tim Langdell

The first section of this book is called 'Getting started on your ZX Spectrum' and that is what you are encouraged to do very soon into the text.

The first section is split into five chapters, first introducing you to the various ins and outs of the ZX Spectrum, and then guiding you through colour and sound, and the other intricacies of BASIC programming. The chapter on graphics in motion provides a simple introduction to the subject which is covered in much more detail in the following section entitled 'Colour, Graphics and Sound'.

The third section of the book is entitled 'Gamesmanship', and introduces the reader to a number of the 'tricks of the trade' with some example games programs to illustrate the points made. There follows a very short chapter on educational and serious applications which includes a small word-processor program, a filing system and a statistics listing.

The final section of the book is called 'Taking your Spectrum to the limits' and it attempts to guide the reader to improving your programming. There is also a chapter on machine code programming and two appendices with the ASCII table and the Spectrum's memory map.

The book is well-illustrated with programs and screen displays, and contains a wealth of hints and tips for the beginner and experienced alike.

The Spectrum Handbook, written by Tim Langdell, is published by Century Publishing Company Ltd. The book has 216 pages and is priced at £4.95.
ISBN 0 7126 0152 X

Exploring Spectrum BASIC — Mike Lord

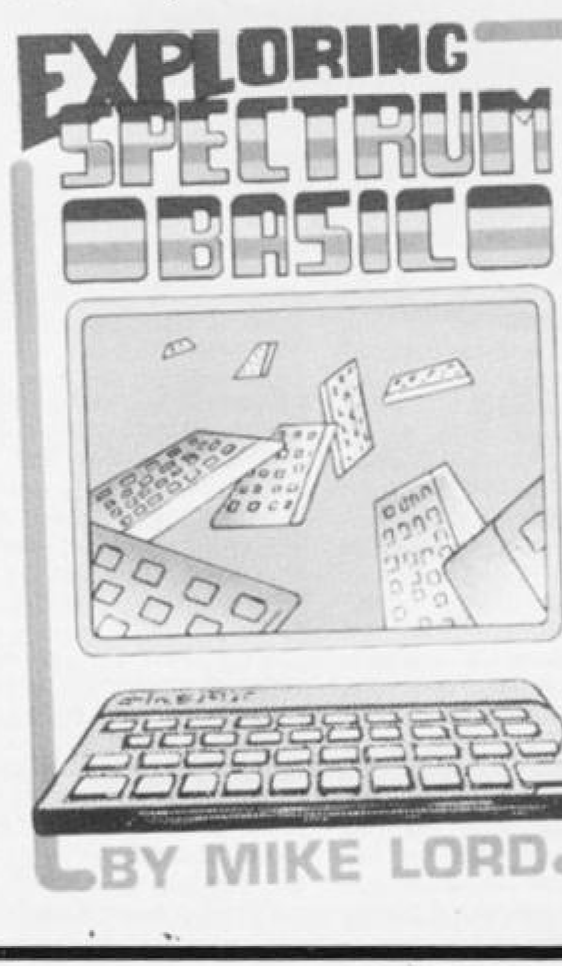
The first six chapters of this book introduce the reader to the ZX Spectrum, the various commands and functions available on the machine, and some of the programming features which can be used such as looping, arrays and data handling. There is also a chapter describing the creation of a program — Calendar — and the subsequent debugging of the listing.

Then, once you have presumably got the hang of manipulating simple programs, you are provided with over 50 full programs together with detailed explanations. As new techniques are used within the programs provided, some explanatory text is given to show the reader what is going on. Numerous short routines are also included within the text for illustrative purposes.

The listings are all clear and easy to read, and the Spectrum keywords are printed in a bold typeface. Screen display dumps are used throughout the book to illustrate the text.

The last two chapters of the book are called 'Applications', containing such programs as Linear Programming, Simultaneous Equations, Pearson's Correlation, and a General Purpose Graph Plotter; and 'Utility programs, quirks, and useful routines', which contains programs to test the ROM and RAM, Renumber routines, etc.

Three appendices are included entitled 'Places to PEEK & POKE'; 'Speeds', which you can use to speed up the rate at which the program runs; and 'Other BASICs', which explains some of the difference between Sinclair BASIC and the rest.



Exploring Spectrum BASIC, written by Mike Lord, is published by Timedata Ltd. The book has 191 pages and is priced at £4.95.
ISBN 0 907892 03 5

Games To Play On Your ZX Spectrum — Martin Wren-Hilton

This book provides the reader with 13 programs for the ZX Spectrum.

Each program comes complete with an explanation of the structure of the listing and a screen photograph of the game once it has been correctly typed in and RUN. The listings are all very easy to read and in some cases, illustrated with quite witty cartoons.

The programs included in this publication are Breakout, String art, Helicopter, Worm race, Flower, Mastermind, Monitor, Bomber, Kaleidoscope, Customer, Spiral, Stunt bike and Draughts.

All the programs will RUN on the standard 16K Spectrum.

Games To Play On Your ZX Spectrum, written by Martin Wren-Hilton, is published by Shiva Publishing Ltd. The book has 43 pages and is priced at £1.95.
ISBN 0 906812 28 3

The Spectrum Programmer — SM Gee

The first two chapters of this book gently introduce you to the idea of learning to use your Spectrum, with some good advice for getting the best out of your machine.

Once set up, consult chapter three which takes you through the first steps of programming, with a look at variables, and the keywords PRINT, LET and INPUT. The following chapters concentrate on looping, handling text and numbers, and using functions and subroutines. Chapter seven introduces you to the use of graphics and completes your introductory study of programming techniques.

It is at the end of chapter seven and the following chapter that the reader is introduced to the idea of using these newly acquired skills to write a simple games program. Chapter eight deals with the use of sound on the Spectrum and ends with a simple program of the alien invader type which illustrates what you should have learnt so far.

Chapter nine deals with high-resolution graphics on the Spectrum showing use of the graphics commands, with sections on high-resolution colours and un-plotting. The final chapter of the book is called 'Logic and other topics', completes the Spectrum tutorial.

The Spectrum Programmer, written by SM Gee, is published by Granada Publishing. The book has 141 pages and is priced at £5.95.
ISBN 0 246 12025 8

Publishers

The books mentioned in this article are published by the following:

Melbourne House (Publishers) Ltd, Glebe Cottage, Glebe House, Station Road, Cheddington, Leighton Buzzard, Bedfordshire LU7 7NA.
Interface, 44-46 Earls Court Road, London W8 6EJ.
Phipps Associates, 99 East Street, Epsom, Surrey KT17 1EA.
Sunshine Books, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF.
Gower Publishing Company Ltd, Gower House, Croft Road, Aldershot, Hampshire GU11 3HR.
Century Publishing Ltd, 76 Old Compton Street, London W1V 5PA.
Timedata Ltd, 16 Hemmels, Laindon, Basildon, Essex.
Shiva Publishing Ltd, 4 Church Lane, Nantwich, Cheshire CW5 5RQ.
Granada Publishing, PO Box 9, Frogmore, St. Albans, Hertfordshire AL2 2NF.



ZX81 16K SPECTRUM RAM Cassettes

Games to Test your Skills & Tactics

"VERY NASTY MOUNTAIN and NASTY INVADERS...are both very well written"; "I am thoroughly pleased with the goods" : Client, London
"Thanks for your very good VERY NASTY MOUNTAIN": M.F. Harold, Guildford, Sy.
"NASTY INVADERS...VERY NASTY MOUNTAIN : with no loading problems at all and the literature you included, will certainly purchase from you again": N.D.H., Huddersfield
"I am writing to congratulate you on the excellent VERY NASTY MOUNTAIN. I am also pleased with the quick delivery": R.Massingham, Batley, W. Yorks.
"I like NASTY INVADERS...the graphics are excellent; VERY NASTY MOUNTAIN is also an excellent game. These are two of the best games I have for my ZX81.": Michael B. McAllister, Lakenheath, Suffolk

NASTY INVADERS: Get them before your Bosses get you! £4.95
A 20 minute plus Action-Packed Game. You are on duty in the Defence Radar Centre. An invasion starts. Your task is to prevent the Enemy from landing. But you have problems: not only do you have to stop the Invaders making repairs to their craft, but there could be personnel trouble too! Good control is rewarded, but errors are penalised.
Don't despair — Rank Has Its Privileges!!

NASTY MOUNTAIN: Solve the Clues if you want to Escape! £4.95
You are leading an expedition and come to an impassable mountain range. It looks like a long detour until an old goat-herd announces that there is an opening in the rocks into which the occasional animal wanders, but they never re-appear..... See if you can solve the mysteries of Nasty Mountain and continue on your travels. Whether you succeed or fail, you can always try again — but somehow its not quite the same as before!

VERY NASTY MOUNTAIN: But don't CHEAT or you will pay for it! £6.95
An advanced version of the Nasty Mountain Game with 16 Levels of Play. Practice Makes Perfect — but the more clues you solve, the more your tactics are tested!

Justify Your ZX 81 As A Business Expense

PETTY CASH/VAT: Know where your Money went! £10.00
Enter Date, Narrative and Gross Expenditure for each Cash Purchase. Analyses into 20 Sub-Headings and 13 Main-Headings and gives Gross, VAT, Net and Exempt Subtotals and Totals. Its so good, we use it ourselves.

VAT BOOK INPUTS: Written for the Small Business & Sole Trader £12.00
As the Petty Cash/VAT Program, but with the facility to add in your Cheque and Standing Order Items to give Inputs Bottom Line Totals.

CHEQUES/P.O.s TO:— GILTROLE LTD., DEPT. ZXC, P.O. BOX 50, RUGBY, WARKS. CV21 4DH

ZX SPECTRUM & ZX81 EDUCATIONAL SOFTWARE

Spectrum Junior Education £5.50
Use your Spectrum to help your children with their school work. This cassette contains eight attractive, easy-to-use programs for the 7 to 11 age group.

Topics include English comprehension, spellings, homophones, junior science, maths and history.
* Entering your own questions and answers allows you to adapt two of these programs for exercises in any subject area.
* Moving colour graphics and sound are extensively used to improve motivation.
* Use the "draw" program to produce pictures, maps and diagrams.

Suitable for the 16K or 48K Spectrum. Program notes are supplied.

O-Level Chemistry (C1) £5.50
This cassette contains four clearly presented revision/tutorial programs. The subject matter has been carefully structured to cover the most important aspects of:

* Elements, compounds and mixtures.
* Structure, bonding and properties.
* Redox, electrolysis and the activity series.
* Acids, bases and salts.

48K Spectrum and 16K ZX81 versions of the cassette are available. Please specify which you require.

Professional Computer Assisted Learning materials from:

CALPAC COMPUTER SOFTWARE
108 Hermitage Woods Crescent, St Johns, Woking, Surrey GU21 1UF.

Overseas orders: £6.50 ($11.60) per cassette; includes AIRMAIL postage.

Great games for your Spectrum



from MIKRO-GEN the leaders in ZX games



Sorcerer's Castle

NEW

Takes you into a world of magic and mystery. Graphical position and a host of options with this exciting adventure game for the 48K Spectrum **£5.50**

Scramble

High-speed arcade game – the fastest available. Thrust, altitude, fire and bomb controls. For 16K or 48K Spectrum. **£5.50**

Masterchess

the program for the *real* player!

Ten levels of play + change sides or level in mid-game + set board to any position + scrolling history + displays your moves and computer's + copy display & history to printer any time + save game at any point + recommend move option + plays all legal moves + indicates illegal moves + Chess Clock on reverse. A powerful, adventurous and enjoyable 48K program **£6.95**

And for your ZX81:

Scramble ★ Frogs ★ PaintMaze ★ ZX Chess ★ Breakout ★ Space Invaders ★ Asteroids ★ Tempest ★ 1K Games-pack ★ Debug

All supplied on cassette with library case £3.95 (ZX Chess £6.50)

Write for full details of the Mikro-Gen range of programs and add-ons, available from local stockists or direct from the manufacturers (please make cheques/PO's payable to Mikro-Gen and add 40p post & packing)

Suppliers of Software to Sinclair

MIKRO-GEN

24 Agar Crescent Bracknell Berks RG12 2BK
Tel: Bracknell (0344) 27317

Home run

Get home as quickly as you can with this program for the 1K ZX81 from Andrew Tierney of Reading.

The object of this game is to move the black square to the chequered square in the shortest possible time.

You move the black square using the '5' key for left, the '6' key for down, the '7' key for up and the '8' key for right. There is a time limit for you to reach your home square; this time limit decreases, thus making your task harder, each time you successfully get home. Your score for each turn is the number of time units you have left when you reach the chequered square.



```
10  LET C = 50
20  LET A = 0
30  LET B = 0
40  LET X = INT (RND * 10)
50  LET Y = INT (RND * 10)
60  PRINT AT X,Y;
    "(Graphic shifted A)"
70  FOR N = 1 TO C
80  PRINT AT A,B;"■"
90  PRINT AT A,B;" "
100 IF INKEY$ = "5" THEN
    LET B = B-1
110 IF INKEY$ = "6" THEN
    LET A=A-1
120 IF INKEYS$ = "7"
    THEN LET A = A+1
130 IF INKEYS$ = "8"
    THEN LET B=B+1
140 IF A=X AND B = Y
    THEN GOTO 180
150 NEXT N
160 PRINT "HARD LUCK"
170 STOP
180 PRINT "WELL DONE -
    SCORE = " C - N
190 PAUSE 50
200 LET C = C-1
210 CLS
220 GOTO 10
```

Note that the contents of the bracket in line 60 should be replaced by the graphics symbol described.

Conquering LOAD/SAVE problems

Tim Hartnell and Ian Beardsmore summarise advice from members of the National ZX Users' Club.



One of the real bugbears of the ZX81 is the touchiness of the cassette interface. While few ZX81 owners have problems getting some sound onto a cassette from their computers, getting this sound back into the ZX81 as a program is often a major hurdle to full enjoyment of the computer.

Some of the following advice will not be relevant to your particular computer/cassette setup, so read through all the advice we give until you find something that appears to apply to you, and they try out the suggestions given.

In a jam?

Make sure the outputs from your cassette player are compatible, that is you have 3.5mm sockets, and check that the plugs fit firmly. Some appear to work better if they are moved a fraction of an inch out of the computer, rather than being jammed hard into the ZX81. The ZX80 and the ZX81 need four to six volts peak output. If you have a DIN socket, it is almost certain not to work. Most DIN outputs are about 1.5V. This is not enough for the ZX81, and will only work if a special buffer circuit is put between the recorder and the computer.

If you have a 16K program, trying SAVEing, then LOADing a very short program without the pack on, and then with it in place. If they both LOAD/SAVE, then try a longer program. If the longer program does not LOAD/SAVE, then the problem could well be overheating, something that can cause problems within 15 minutes of turning the computer on.

If you get LOAD/SAVE problems only with the 16K RAM Pack attached, then try the following ideas. They sometimes bring results, even with mischievous RAM packs.

1. Keep your ZX cool. This can be done by using a fan, or by placing something cold on top. Tim's first book, **Making the most of your ZX80,** was written with the aid of a succession of frozen Long Life milk cartons. If you follow this somewhat bizarre idea, make sure that the water doesn't get into the computer. If you have an extreme heating problem, and you feel confident of your ability to do it, you can take the top off, and/or solder in an extra heat sink.
2. Don't jar the computer while SAVEing or LOADing.
3. Try different volume levels when LOADing, and keep the tone control on maximum (ie full treble). You'll need a lot of volume in most cases, and maximum treble.
4. Noise can be a source of LOADing problems. There should be about five seconds of silence before a program. Tape hiss is not conducive to a successful load.
5. Use good, proper computer cassettes, rather than cheap low-noise audio ones. Quality audio cassettes (such as TDK) generally work well.
6. Clean the tape head frequently, and — if you can get one — use a demagnetiser from time to time.
7. If your cassette recorder has batteries, try it with them, rather than running the recorder from the mains. You should find this helps.
8. Make several copies of each program, so if one doesn't load, you can always try the next one on the tape. Frequent use of a tape (or one portion of it) will lead to a buildup of hiss over the program, which may eventually make it impossible to load from.

Hardware aids

The following suggestions are for you if you can solder confidently. These suggestions are more for ways of checking that the signal is getting from the recorder to the computer, than specific aids in LOAD/SAVE.

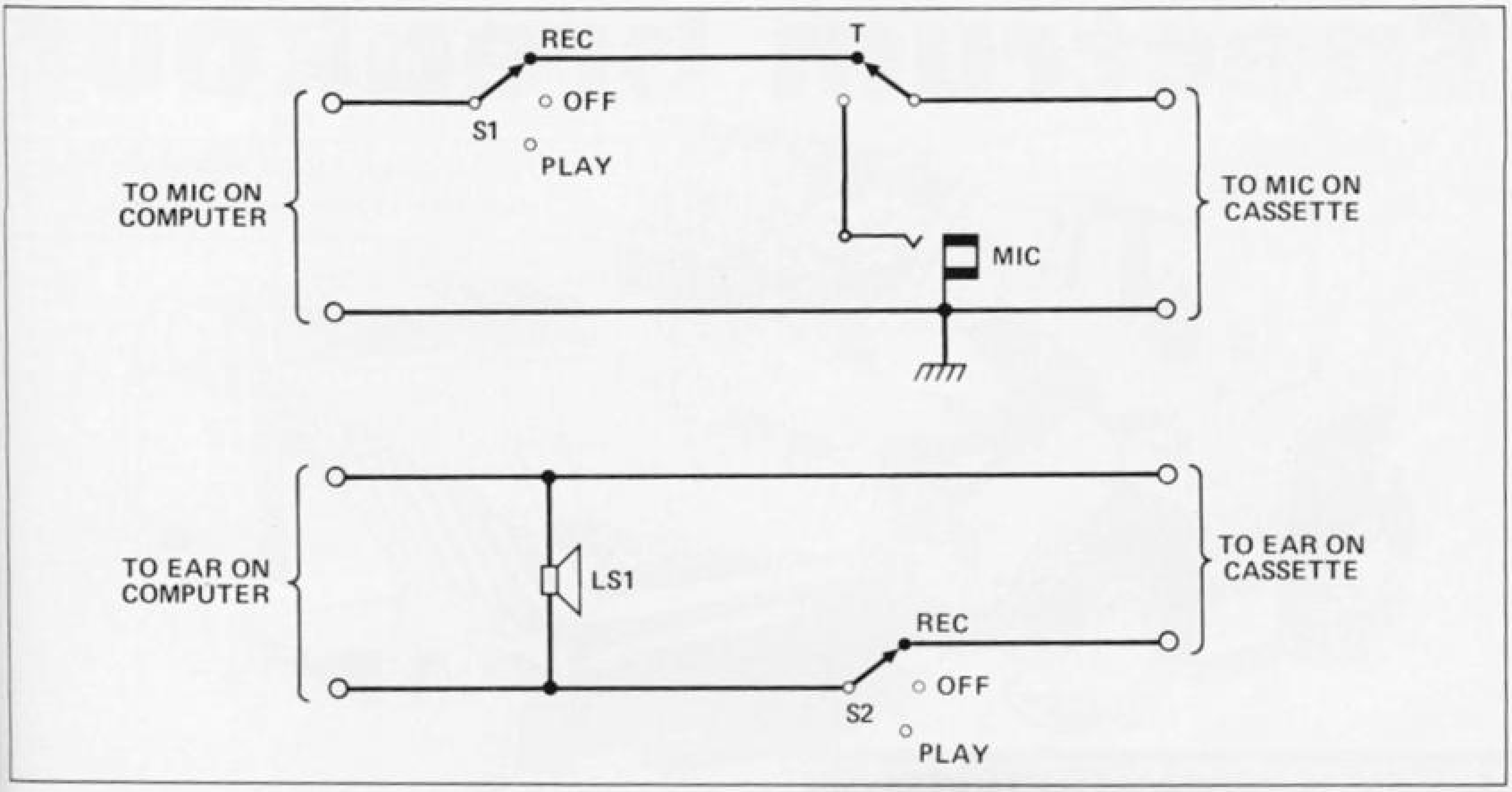
1. Wire an LED into the leads.
2. Use crocodile clips to tap directly off the loudspeaker.
3. Connect a wire across the spring-loaded loudspeaker/earphone switch (in the recorder across the earpiece jack), so you can hear what is going on.
4. Wire a small earpiece into the leads.
5. Wire the loading lead directly from the output signal. Fit a switch which can be used to disconnect the internal speaker while loading.

Which recorder?

The Users' Club has often been asked 'Which cassette recorder works best?'. We cannot endorse a particular brand of recorder, and do not suggest that while club members have been successful with the following brands, you will necessarily have the same success, but for what it's worth, here are some of the recorders which club members have used successfully: ITT Studio recorder 66; Boots CTR500; Tandy Micro minisette II; Sony TCP55; Interstate/Waltham (Woolworths); Hitachi TRQ291 and TRQ247; Prinzsound TR2256; Ferguson 327; Sanyo M2406P; Prinz SC9; and the machine which is widely advertised for use with the ZX81, the Monolith ECR81, data-assette E312.

Adjusting azimuth

Mr S Atkinson of Harrogate suggests the following. He points out that, for loading, the output from the cassette needs to contain as much treble as possible. You can increase the treble content of the recorder's output by changing the tape head azimuth, the alignment between the tape head and the tape.



The tape head on most cassette machines is mounted on two screws, one of which is sprung. By adjusting this screw, the tone can be altered. The screw is accessed by a small hole above the play head. In many cases, this hole is covered by tape, or a small metal plate.

Here's how you can adjust your azimuth:

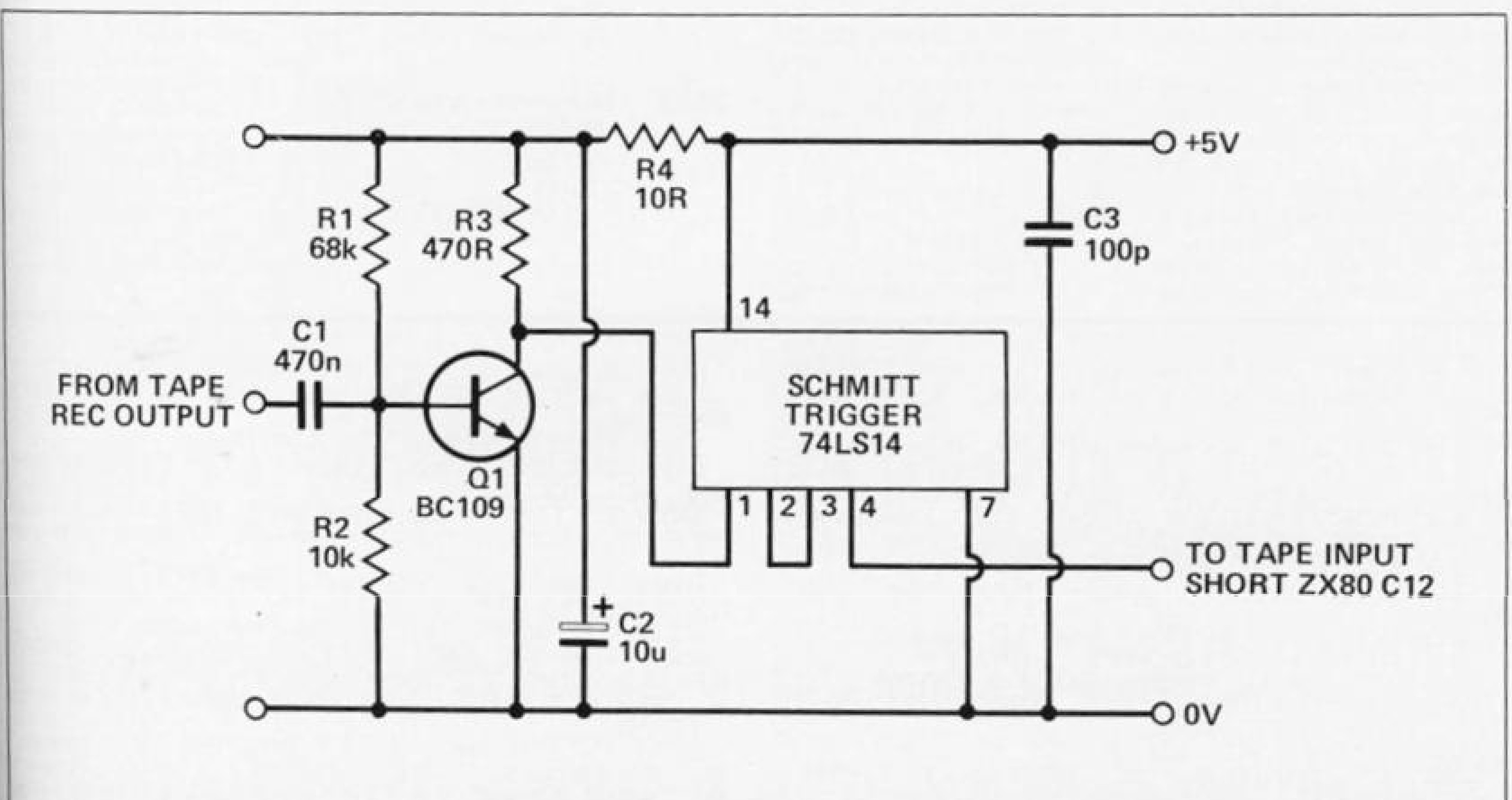
1. Turn the recorder controls to maximum treble, minimum bass, and insert a tape containing a program.
2. With the earpiece out, start the recorder.
3. Adjust the screw, trying it in both directions, until the sound is as 'tinny' as possible.

Note that this may make some of your music cassettes sound a little too sharp for your liking.

Fig. 1. S1 and S2 are both 2-pole, 3-way slide switches and S3 is a 1-pole, 2-way biased push button slide switch. J is a 3.5mm jack socket for a microphone and LS is a miniature 75Ohm loudspeaker. You will also need four 3.5mm jack plugs with screened leads. This circuit can be contained in a box measuring 3" by 2½" by 1½".

The switching allows recording and playback without removing plugs. The push button allows audio inject for identification. The speaker is used for audio monitoring.

Fig. 2. This circuit is for use with low level output from the tape deck. Adjust R1 and R2 for different outputs from the tape deck. This circuit was designed to operate with a Sanyo 5050G deck, 580mV at 5.6K output.



Spectrum breakout



See page 114 for further details

Here's a program written by David Spencer of Spalding for all you Spectrum owners who fancy a spot of demolition.

Based on the 16K ZX Spectrum, this game takes just over 6K of memory and makes full use of the sound and colour available on the machine.

The wall itself comprises three different coloured layers, each with 30 bricks. When 60 bricks have been knocked out, a new wall appears and the scores for each layer increases.

When you begin to play the game, you have five lives. However, each time you miss the ball with your bat, you will lose a life. The game ends when you have cleared all five walls or you have lost all five of your lives. All the appropriate sound effects have been included for when you hit the ball with your bat and when you knock out a brick from the wall.

When the game is not being played, the program cycles through a sequence of two explanation pages followed by a demonstration game in which the computer guides the bat. The highest score and the relevant player's name is displayed on screen at this time.

Here is a list of all the variables used in the program.

HS	– Highest score.
N$	– Name of the highest scorer.
F	– General FOR . . . NEXT counter.
FP	– Play flag (1 for the demonstration game).
P	– X co-ordinate of the bat.
W	– Wall number.
S	– Score.
ES	– Temporary INPUT string.
X,Y	– Co-ordinates of the ball.
DX,DY	– Movement vector for the ball.
FL	– Flag for sound effects.
FT	– Flag for hitting top of screen.
FA	– Flag for hitting bat.
L	– Lives.
A	– Variable to indicate which layer has been hit.
J	– Set to 0 for game, 8 for demonstration game.
C	– Code of character at ball position.
H	– Number of bricks knocked out.
P1	– Required position of bat during demonstration game.
DP	– Direction of bat movement during demonstration game.
YT	– Temporary variable used in determining P1.

```
   1 REM **BREAK-OUT
   3 BORDER 0: LET hs=0: LET n$=
""
   5 GO SUB 2000: PRINT AT 21,2;
  10 PRINT AT 19,5; FLASH 1; BRI
GHT 1;"PRESS ANY KEY TO PLAY"
  12 FOR f=1 TO 1600: IF INKEY$<
>"" THEN GO TO 15
  13 NEXT f: GO TO 3000
  15 BEEP .5,0: LET fp=0: LET j=
0
  20 GO SUB 9000: GO TO 1000
  99 REM **KEYBOARD SCAN**
 100 IF fp=1 THEN GO TO 5000
 105 IF INKEY$<>"z" AND INKEY$<>
"m" THEN RETURN
 110 PRINT AT 21,p;"   "
 120 IF INKEY$="z" AND p>1 THEN
LET p=p-1
 130 IF INKEY$="m" AND p<29 THEN
LET p=p+1
 140 PRINT AT 21,p; INK 2;"■■■":
RETURN
 199 REM **END OF WALL**
 200 PAUSE 100: CLS : IF fp=1 TH
EN GO TO 5
 203 LET w=w+1: PRINT AT 4,4; PA
PER 5;"Wall no. ";w-1;" destroye
d."
 205 FOR f=0 TO 10: BEEP .2,f: N
EXT f
 210 IF w=6 THEN GO TO 230
 220 PAUSE 50: PRINT AT 6,4; PAP
ER 4; INK 7;"Now go onto wall ";
w: PAUSE 150: GO SUB 9007: GO TO
 1000
 230 PRINT AT 6,3; PAPER 4; INK
0; FLASH 1;"WELL DONE, "; FLASH
0;"you have destroyed all the wa
```

```
lls"
 240 PRINT AT 9,4;"Your score wa
s ";s
 249 REM **HIGH. SCORE ROUTINE**
 250 IF s<hs THEN PRINT AT 11,4;
"Therefore you did not beat  the
 highest score of ";hs;" by ";n$
: GO TO 600
 260 IF s=hs THEN PRINT AT 11,4;
"And you equaled the highest sco
re": INPUT "ENTER YOUR NAME "; L
INE e$: IF e$="" THEN GO TO 260
 265 IF s=hs THEN LET n$=n$+" &
"+e$: GO TO 600
 270 IF s>hs THEN PRINT AT 11,4;
"And you beat the highest    sco
re!": INPUT "ENTER YOUR NAME ";
LINE n$: IF n$="" THEN GO TO 270
 280 LET hs=s: GO TO 600
 299 REM **ALL LIVES LOST**
 300 FOR f=0 TO 7: PAPER f: CLS
: PAUSE 20: NEXT f
 310 PRINT AT 4,11; INVERSE 1;"G
AME OVER": PAUSE 50: GO TO 240
 600 FOR f=15 TO -5 STEP -1: BEE
P .3,f: NEXT f
 610 PAUSE 200: GO TO 5
 999 REM **MAIN BALL ROUTINE**
1000 PRINT AT y,x;"o"
1002 IF fp=1 THEN GO SUB 5050
1003 IF fl=1 AND fp=0 THEN BEEP
.05,a*4: LET fl=0
1005 GO SUB 100: LET y=y+dy: LET
 x=x+dx: IF x=1 OR x=30 THEN LET
 dx=-dx: PRINT AT y-dy,x+dx;" ":
 GO TO 1009
1006 IF ft=1 THEN LET ft=0: GO T
O 1009
1007 PRINT AT y-dy,x-dx;" ": IF
fa=1 THEN GO SUB 140: LET fa=0
1008 IF fp=1 THEN GO SUB 5100
1010 IF y>6 AND y<21 THEN GO TO
1000
1020 IF y<7 THEN GO TO 1050
1029 REM **HIT BAT ROUTINE**
1030 IF x=p OR x=p+1 THEN LET dy
=-dy: LET x=x+1-(2*(x>=29)): LET
 y=y-1: LET fa=1: BEEP .1*(fp=0)
,0: GO TO 1005
1035 IF INKEY$<>"" THEN GO TO 10
35
1040 BEEP .75*(fp=0),30: LET l=l
-1: PAUSE 50: IF l=0 THEN GO TO
300
1045 PRINT AT 21,p;"  ": GO SUB
9050: GO TO 1000
1049 REM **HIT WALL ROUTINE**.
1050: LET a=ATTR (y,x)-48-j: LET
 c=CODE SCREEN$ (y,x): IF a=1 AN
D y>1 AND c<>95 THEN GO TO 1000
1055 IF a=1 AND c<>95 THEN LET d
y=-dy: GO TO 1000
1070 IF c=95 THEN GO TO 1100
1075 LET fl=1
1080 LET s=s+(((a-2)*10)*w): LET
 h=h+1: PRINT AT 0,9;s: IF h=60
THEN GO TO 200
1090 LET dy=-dy: GO TO 1000
1100 LET x=x+(2*(RND>.5)-1): LET
 dy=-dy: LET ft=1: GO TO 1005
1999 REM **1st EXP. PAGE**
2000 PAPER 7: CLS : INK 0
2010 PRINT AT 2,10; PAPER 6;"BRE
AK-OUT";AT 2,9; INK 2; OVER 1;"_
___________"
2020 PRINT AT 4,0; PAPER 6;"
 The object of the game is  to h
it bricks out of the wall byboun
cing the ball off the bat.     "
2030 PRINT AT 9,0; PAPER 4;"
When you have knocked out 60 bri
cks a new wall appears. The game
 ends after 5 walls,or when you
runout of lives.               "
2040 PRINT AT 15,0; PAPER 2;"
 You move the bat left with  the
 Z key,and right with the M.  "
2050 RETURN
2999 REM **2nd EXP. PAGE**
3000 CLS : PRINT AT 2,10; PAPER
6;"BREAK-OUT";AT 2,9; INK 2; OVE
R 1;"___________"
3010 PRINT AT 4,1; PAPER 4;"
You start with 5 lives        "
3020 PRINT AT 6,1; PAPER 5;"
The highest score is ";hs;"by "
;n$;".";AT 7,0; PAPER 7;"  "
3030 PRINT AT 9,1; PAPER 6;" On
the first wall the scores ";AT 1
0,1; PAPER 6;"are as follows:-"
3040 PRINT AT 11,7; PAPER 6;"10
for the bottom layer.";AT 12,7;"
20 for the middle layer.";AT 13,
7;"30 for the top layer.   "
3050 PRINT AT 15,1; PAPER 3;" On
 the next wall the scores  ";AT
16,1;"are twice the above and so
 on."
3060 PRINT AT 19,5; FLASH 1; BRI
GHT 1;"PRESS ANY KEY TO PLAY"
3070 PRINT AT 21,2; INK 3;"© Jul
y 1982, David M. Spencer."
3080 FOR f=1 TO 1600: IF INKEY$<
>"" THEN GO TO 15
3090 NEXT f
3999 REM **DEMO. GAME**
4000 LET j=8: LET fp=1: LET p1=0
: LET dp=0: GO SUB 9000: GO TO 1
000
4100 IF x<15 AND dx=1 THEN LET p
1=x+14: GO TO 4140
4105 IF x>15 AND dx=-1 THEN LET
p1=x-14: GO TO 4140
4110 IF dx=1 THEN LET yt=7+(32-x
): LET p1=30-(22-yt)
4120 IF dx=-1 THEN LET yt=7+x: L
ET p1=22-yt
4140 IF p1=0 THEN LET p1=1
4145 IF p1>=30 THEN LET p1=29
4150 IF p1=p THEN LET dp=0: RETU
RN
4160 IF p>p1 THEN LET dp=-1: RET
URN
4170 LET dp=1: RETURN
4999 REM **AUTO BAT MOVE**
5000 IF INKEY$<>"" THEN GO TO 15
5010 IF p=p1 OR p+1=p1 THEN RETU
RN
5020 PRINT AT 21,p;"  ": LET p=p
+dp: GO TO 140
5050 IF dy=-1 THEN LET p1=15: GO
 TO 4150
5060 RETURN
5100 IF y=7 AND dy=1 THEN GO TO
4100
5110 RETURN
8999 REM **INIT. ROUTINE**
9000 LET s=0: LET w=1: LET l=5
9007 LET h=0
9010 BORDER 0: PAPER 6: INK 1: C
LS
9015 IF fp=1 THEN PAPER 7: CLS
9020 FOR f=0 TO 21: PRINT PAPER
0;AT f,0;"█";AT f,31;"█": NEXT f
9030 FOR f=1 TO 3: PRINT AT 1,f*
7;"_______": NEXT f
9040 FOR f=1 TO 30: PRINT INK 5;
AT 4,f;CHR$ 143; INK 4;AT 5,f;CH
R$ 143; INK 3;AT 6,f;CHR$ 143: N
EXT f
9050 PRINT AT 0,3;"Score ";s;AT
0,22;"Lives ";l
9055 IF fp=1 THEN GO TO 9080
9060 PRINT AT 18,5; FLASH 1; PAP
ER 0; INK 5;"PRESS ANY KEY TO ST
ART"
9070 PAUSE 0: PRINT AT 18,5; PAP
ER 6;"                      "
9080 LET p=15: LET x=INT (RND*28
)+2: LET y=20: LET dx=-1+(2*(RND
>.5)): LET dy=-1: LET fa=0: LET
p1=0: LET ft=0: GO TO 140
```



*SPECTRUM MONITOR *SPECTRUM EDITOR/ASSEMBLER *ZX81 SCREEN KIT 1 *ZX81 ZX-MC *ZX81 REMLOAD

FAST MAIL ORDER-SEND TODAY!

SPECTRUM MONITOR. Machine Code Debug/Disassembler
• Enter, Run, Debug machine code programs. • Compatible with Basic • Breakpoints and Registers Display • Disassembly to screen and/or ZX Printer • 16K and 48K versions on one cassette + 30 page manual. **£7.50**

SPECTRUM EDITOR/ASSEMBLER. A powerful and essential machine code programming aid. 16K & 48K on same cassette with full documentation. Major features include:
EDITOR with Auto Line Numbering. 40 Column screen display, tabulated into fields for easy reading. 5 character Label Names: simple Line Editing and Cursor Control: SAVE/LOAD Text Buffer to cassette: output to ZX PRINTER.
TWO-PASS ASSEMBLER accepts all Z80 mnemonics (plus many unpublished mnemonics). Decimal or Hex numbers: simple arithmetic on operands. Assembler Directives — ORG, END, DEFB, DEFW, DEFS, DEFL, EQU, DEFM.
WE CANNOT FULLY DESCRIBE THIS IMPORTANT UTILITY HERE, AND ASK YOU TO SEND A S A E. FOR COMPLETE DETAILS OF THIS AND ALL OUR PROGRAMS. **£8.50**

ZX81 SCREEN KIT 1. More power to your screen in all your Basic programs
BORDERS any size, anywhere on screen. SCROLL in all 4 directions. CLEAR and REVERSE PART OF SCREEN. FLASHING CURSOR anywhere on screen — simulates INPUT. DATA FILES SAVE AND LOAD Basic variables. Double Speed. 880 bytes machine code for INSTANT RESPONSE. Becomes part of Basic Program. **£5.70** 4K to 64K

ZX81 ZX-MC. Machine Code Debug/Monitor
• ENTER, RUN, DEBUG machine code programs • SAVE, LOAD, VERIFY at double speed • BREAKPOINTS and REGISTERS DISPLAY • Self contained — cannot be used with Basic • Cassette plus 36 page manual. **£7.50** 4K to 64K

ZX81 REMLOAD. Machine Code Debug/Monitor
• A version of ZX-MC without the Save/Load/Verify facility • Compatible with Basic • CREATE A REM LINE of any length • BREAKPOINTS and REGISTERS DISPLAY • Cassette plus 30 page manual. **£6.95** 16K to 64K

SEND S.A.E. NOW FOR DETAILS
6 Corkscrew Hill, West Wickham, Kent BR4 9BB.

PICTURESQUE

Prices include VAT & P&P

MICROSPHERE

MICROSPHERE COMPUTER SERVICES LTD
72 ROSEBERY ROAD · LONDON N10 2LA
TELEPHONE 01-883 9411

OMNICALC (48K Spectrum) - The very best spreadsheet analysis program you can buy for the Spectrum. Uses include financial planning, home budgets etc. - in fact any work with figures.
Up to 99 columns or 250 rows * supports ALL the mathematical functions * All input is prompted * Separate headings always available so you know precisely where you are! * Extensive repeat facilities reduce the time spent in setting up models Allows conditional expressions (for tax bands etc.) and RND (for simulations) * Totalling and sub-totalling facilities * Complete with comprehensive manual. £9.95

ZX-SIDEPRINT (Spectrum and ZX81 versions) - Don't be limited to 32 characters on a line with your Sinclair printer. Using ZX-SIDEPRINT you can print sideways to give any line-width. It is easy to add to existing programs, includes routines to take print lines direct from the screen, and can even print in lower case from a ZX81! (State version rqd) £4.95

CREVASSE + HOTFOOT (Spectrum 16K) - Two original games using the Spectrum's abilities to the full. In CREVASSE you are on the ice-cap with chasms opening and closing all around you. Can you make it across the perilous ice? In HOTFOOT you must help a rabbit get to the carrots in the fields around its burrow. But the fields are dangerous, and the plumper the rabbit the greater the danger. The only safety (and the only way to score!) is to get back to the burrow. Now............where was it? £4.95

EVOLUTION (48K Spectrum) - Starting from the primaeval soup, can you retrace the course of evolution and end up with man? Can you keep the earth's eco-systems in balance, respond to climatic changes and survive the odd cataclysm? It should be easy ... after all it has been done before! £6.95

CASSETTE LABELS - If you write programs yourself, or want to find a special gift for someone who does, why not have your own cassette labels printed complete with logo? Prices start at only £4.95 for 50 labels or £7.95 for five C20 cassettes with labels fitted. Send s.a.e for design and order form.

NEW

SAVE IT!

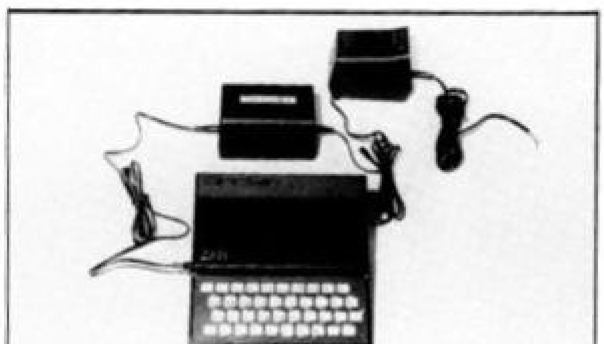
FOR ZX 81 AND SPECTRUM COMPUTERS

(AND OTHER MODELS USING 9v DC ADAPTORS)

with battpack ®

(Patent applied for)

Only **£13·95** + £1·40 p/p



PROTECT YOUR PROGRAMMES WITH THIS RECHARGEABLE BATTERY PACK

AT LAST! The answer to one of the major problems facing personal computer programmers and users: — **POOR MAINS SUPPLIES CAUSING HAVOC, WHILE YOU SPEND HOURS TRYING TO ENTER YOUR CAREFULLY PREPARED PROGRAMMES.**

Now Adaptors and Eliminators, who have produced nearly 70% of all ZX Mains Adaptors, bring you **BATTPACK** — a rechargable 9v Ni-Cad battery pack and transient suppressor, housed in a complementary black ABS case that simply plugs into your ZX adaptor in seconds.

BATTPACK ensures the voltage into your ZX computer never drops below 9v, thereby eliminating the possibility of a programme crash because of a split second mains dropout! (One cycle is all it needs!)

BATTPACK also suppresses transients and RFI using carefully selected capacitors. Your ZX Adaptor keeps the **BATTPACK** at full charge even when in use — so its always ready for emergencies. Even in the event of a full power cut **BATTPACK** will run your ZX computer for at least 30 minutes giving you time to save your programme on a cassette or print a hard copy on your ZX printer.

BATTPACK comes complete with all connecting leads and is ready for use.

ORDER NOW — SAVE NOT ONLY YOUR PROGRAMMES BUT ALSO YOUR TEMPER!!

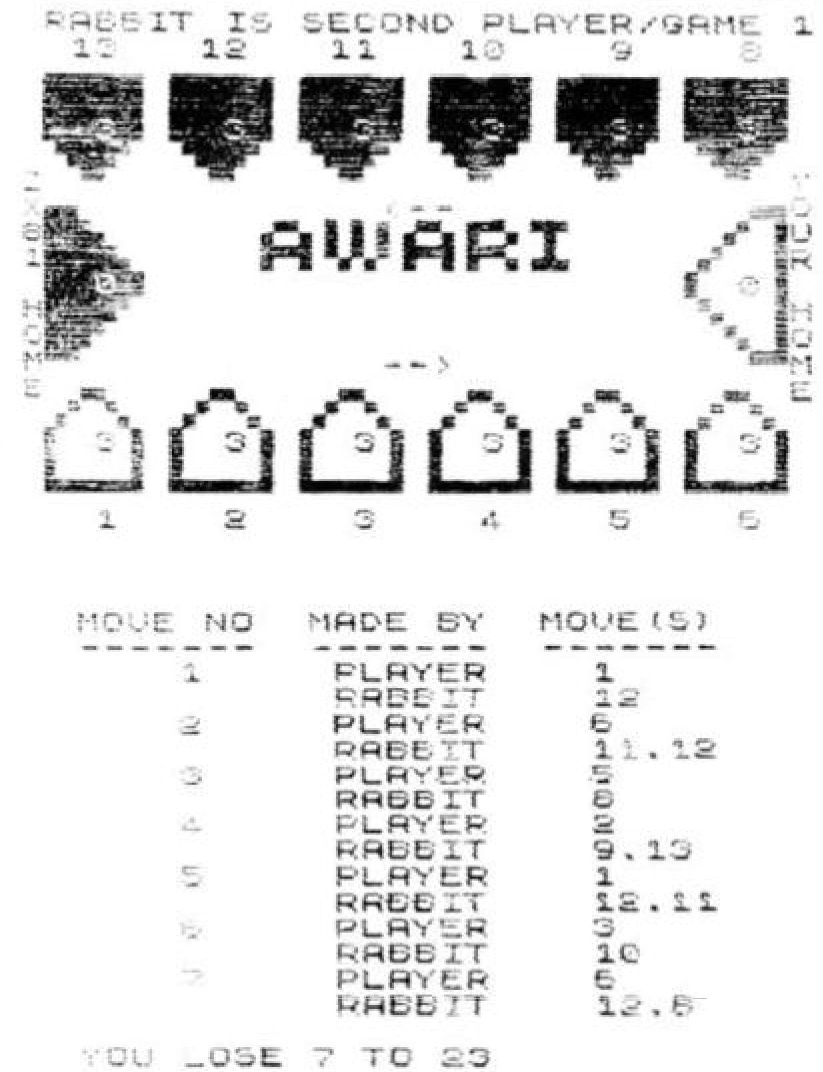
MAIL ORDER ONLY. PLEASE ALLOW 28 DAYS FOR DELIVERY.

ADAPTORS AND ELIMINATORS LTD 14 THAMES ST LOUTH LINCS.
PLEASE SUPPLY_______(QTY) battpack/s
AT £13·95 each PLUS £1·40 p/p
I ENCLOSE CHEQUE POSTAL ORDER FOR £_______

NAME
ADDRESS

In the air tonight

After a brief time in simulated flight, James Walsh comes down to earth to look at the African game, Awari.



I've got two programs to look at for the 16K ZX81, both of which are a little different from the run of the mill stuff. First, we have Awari (no, this is not a mis-spelling of a certain well-known computer company) which is available from Understanding Limited (they must have LOADing problems too!!) for £5.95. There are various prizes to be won depending on where you buy it. The second program is called Pilot, which is available from the people at Hewson Consultants for £5.95.

I shall start with Awari, which is an ancient African game of logic. The game is played using 14 'bowls' each of which may hold any number of 'beans'. Bowls numbered 1 to 6 belong to you, ie you can move the beans out of them, and your 'home' bowl is number 7 on the right. Your objective is to get as many beans into bowl 7 as possible and so to beat the computer. Similarly, bowls 8 to 13 at the top 'belong' to the ZX81, and bowl 14 on the left is its 'home'.

The idea of the game is that you may choose one of the bowls numbered from 1 to 6, which contain at least 1 bean, remove them and place them one at a time in the bowls to the right and anti-clockwise. The ZX81 then chooses from bowls 8 to 13 and moves to the left and anti-clockwise. When you get more proficient, you can investigate the more complicated moves. There are three levels of play, which will accommodate an intelligent eight year old to an adult.

Though the game is basically quite simple, it is more addictive than most non-high-speed games are. The game is well set out and has a good amount of intelligently used graphics.

In conclusion, though it is not the most sophisticated ZX81 program on the market, Awari is ideal for all the family. By the way, it LOADed first time.

Awari is priced at £5.95 and is available from Understanding Ltd, The Production Village, 100 Cricklewood Lane, London NW2 2D5.

Come fly with me

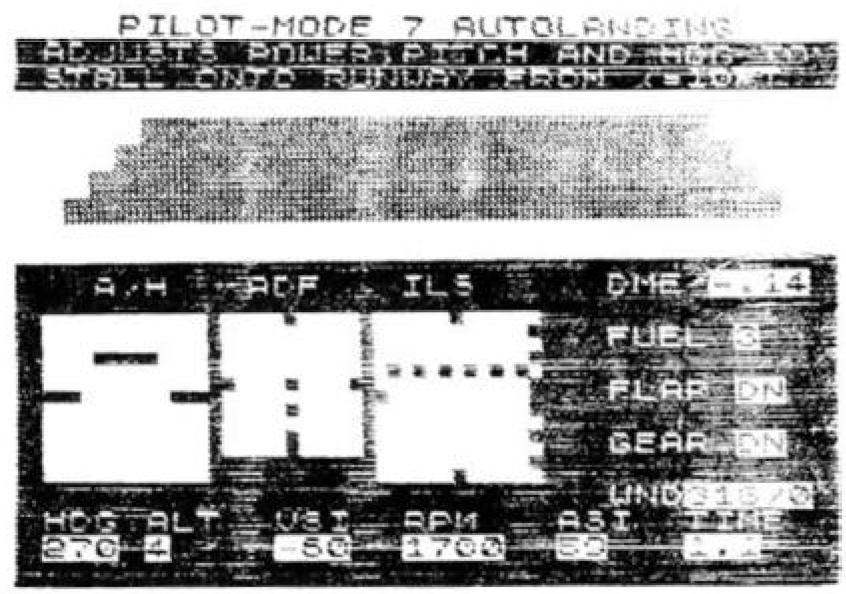
Pilot is a new flight simulation program from Hewson Consultants, written for the ZX81. There is also a ZX Spectrum version available for the same price of £5.95, which is fundamentally the same but with graphics more suited to the capabilities of the Spectrum.

The idea behind a flight simulation program is that you are piloting your own aircraft and are in total control over it. First, you can take off using the navigation beacons, maneouver your way through mountains using the detailed direction finding equipment, and while battling against the forever-changing tropospheric conditions, you must land using manual control or autopilot. Never is life made easy for you, with an astounding, and at times confusing, array of dials, counters, and controls. Unlike most other flight simulation programs for the ZX81, you don't just have the standard altitude, distance and speed commands, but an array including: Artificial Horizon, Automatic Direction Finder, Instrument Landing System, Wind direction and speed, heading, VHF Omni Directional Range...and the list goes on.

The game is definitely not easy and warrents good and extensive instructions. The actual instructions are available in sufficient quantity, but they are not particularly exciting or easy to understand. This was not written for the bloke who wants the next step up from *'Space Invaders' or 'Pacman'*, but it is more of a thinking person's game. I was surprised by the statement in the instructions which read: 'This program is not a game'. This is a strange thing to say as it fulfills all the attributes which normally are assigned to the word 'game', in that it is a contest in which skill is required and is done as a pastime, rather than as work.

The program is very well written and the graphics are pretty good for the ZX81, though some people may get bogged down with the instructions. If you are interested in this type of game, then I can recommend it as the best one that I have seen on the market for the ZX81 so far.

Pilot is priced at £5.95 and is available from Hewson Consultants, 60a St. Mary's Street, Wallingford, Oxon OX10 OEL.



Mastermind '80



Ian Turtle presents us with a ZX80 version of the popular game.

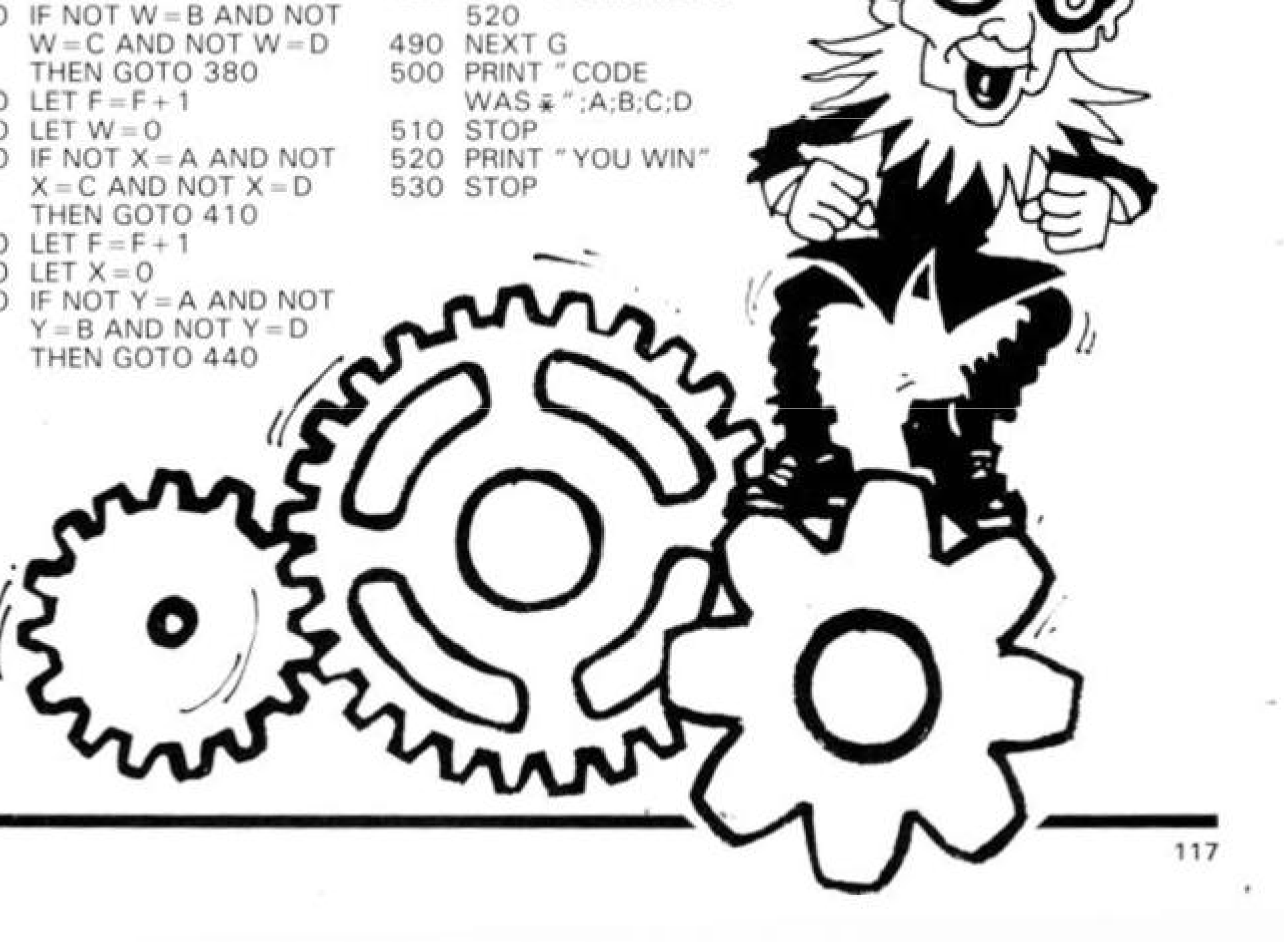
The computer first selects four different random numbers from the limits you suggest — you then have ten rounds in which to guess the hidden code.

Each round consists of you inputting your guess by typing each number and pressing Newline. As the ZX80 accepts each each number, your guess will be displayed on the screen. The number of black and white pegs you are awarded is then worked out once you have guessed four numbers; a black peg is awarded for each number correctly placed in the code, a white peg for each correct number but wrongly placed. Your aim in the game is to achieve four black pegs thus signifying four correct numbers in the right order.

There is a display of all your previous guesses and the black and white pegs awarded, to help you make your next decision.

Note that the symbol '✱' is used in the listing to indicate a space where it is not obvious (usually in PRINT statements before semi-colons).

```
20  RANDOMIZE
30  PRINT "HOW MANY
    NOS."
40  INPUT N
50  IF N<4 THEN GOTO 40
60  LET A=RND(N)
70  LET B=RND(N)
80  IF A=B THEN GOTO 70
90  LET C=RND(N)
100 IF C=B OR C=A THEN
    GOTO 90
110 LET D=RND(N)
120 IF D=C OR D=B OR
    D=A THEN GOTO 110
130 CLS
135 PRINT "GUESS",
    "✱✱CODE",
    "✱✱B✱–✱W"
140 FOR G=1 TO 10
150 LET E=0
160 LET F=0
170 PRINT "✱";G;".",
180 INPUT W
185 PRINT W;"✱";
190 INPUT X
195 PRINT X;"✱";
200 INPUT Y
205 PRINT Y;"✱";
210 INPUT Z
215 PRINT Z;"✱✱";
230 IF NOT W=A THEN
    GOTO 260
240 LET E=E+1
250 LET W=0
260 IF NOT X=B THEN
    GOTO 290
270 LET E=E+1
280 LET X=0
290 IF NOT Y=C THEN
    GOTO 320
300 LET E=E+1
310 LET Y=0
320 IF NOT Z=D THEN
    GOTO 350
330 LET E=E+1
340 LET Z=0
350 IF NOT W=B AND NOT
    W=C AND NOT W=D
    THEN GOTO 380
360 LET F=F+1
370 LET W=0
380 IF NOT X=A AND NOT
    X=C AND NOT X=D
    THEN GOTO 410
390 LET F=F+1
400 LET X=0
410 IF NOT Y=A AND NOT
    Y=B AND NOT Y=D
    THEN GOTO 440
420 LET F=F+1
430 LET Y=0
440 IF NOT Z=A AND NOT
    Z=B AND NOT Z=C
    THEN GOTO 470
450 LET F=F+1
460 LET Z=0
470 PRINT E;"✱–✱";F
480 IF E=4 THEN GOTO
    520
490 NEXT G
500 PRINT "CODE
    WAS✱";A;B;C;D
510 STOP
520 PRINT "YOU WIN"
530 STOP
```



Spectrum maze

Bill Longley of Colchester has written a program for you to lose yourself in your 16K Spectrum.

The game's object is to get out of the maze in the shortest time. This is not quite as simple as it sounds as you can only see a quarter of the maze at any one time, and the whole thing is four screens long rather than the usual one. The way this is done is by moving the maze to the left or right rather than moving you. Normally, this would be very slow, so the movement is done by machine code. So is the actual printing of the maze, and this means there is approximately an eighth of a kilobyte of machine code.

Changing a part of the maze is very easy. Each DATA statement from 2000 to 2117 controls one column on the screen. The screen is divided into three groups of eight lines. Then, each square in the column is coded into a binary digit – a black square is one, a white square is zero. Finally, each group of eight squares is converted into a decimal number between zero and 255, and the three numbers derived like this are entered into the DATA statement. Have a look at Fig. 1 if you still don't understand.

Four of the routines which are in machine code can be used in your own programs. These are as follows:

Line 1: Moves the colours one character to the left. The starting address is 23760.
Line 2: Moves the colours one character to the right. The starting address is 23798.
Line 3: Smoothly moves the actual characters to the left by one square (scrolls by one pixel each time, for eight times). The starting address is 23836.
Line 4: Smoothly moves the characters to the right by one square (scrolls by the same methods as used in line 3). The starting address is 23876.

The fifth line is also machine code, but is of no use outside this program. If these routines are all that are wanted, just enter lines 1 to 4, 6 to 9, and 100 to 460. RUN them, then delete all the lines from 6 onwards. (The machine code will stay in place in the REM.)

How to play

The game uses the four cursor keys in the usual way. Setting up the screen takes about thirty seconds, then you appear as the letter 'A'. You can programme this to be any shape you want. Be careful of memory when you make any additions to the program – I used most of it for the maze. You have about 1K left; however, someone with the 48K Spectrum though, can make the maze up to 32 screens in length! But as this means that over a thousand DATA statements have to be typed in, you'll need strong fingers.

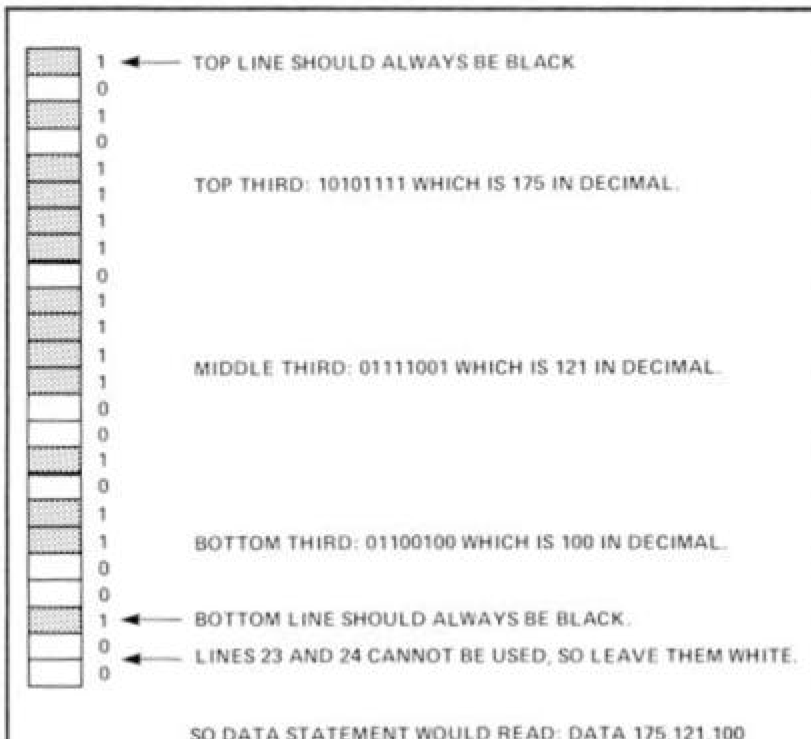


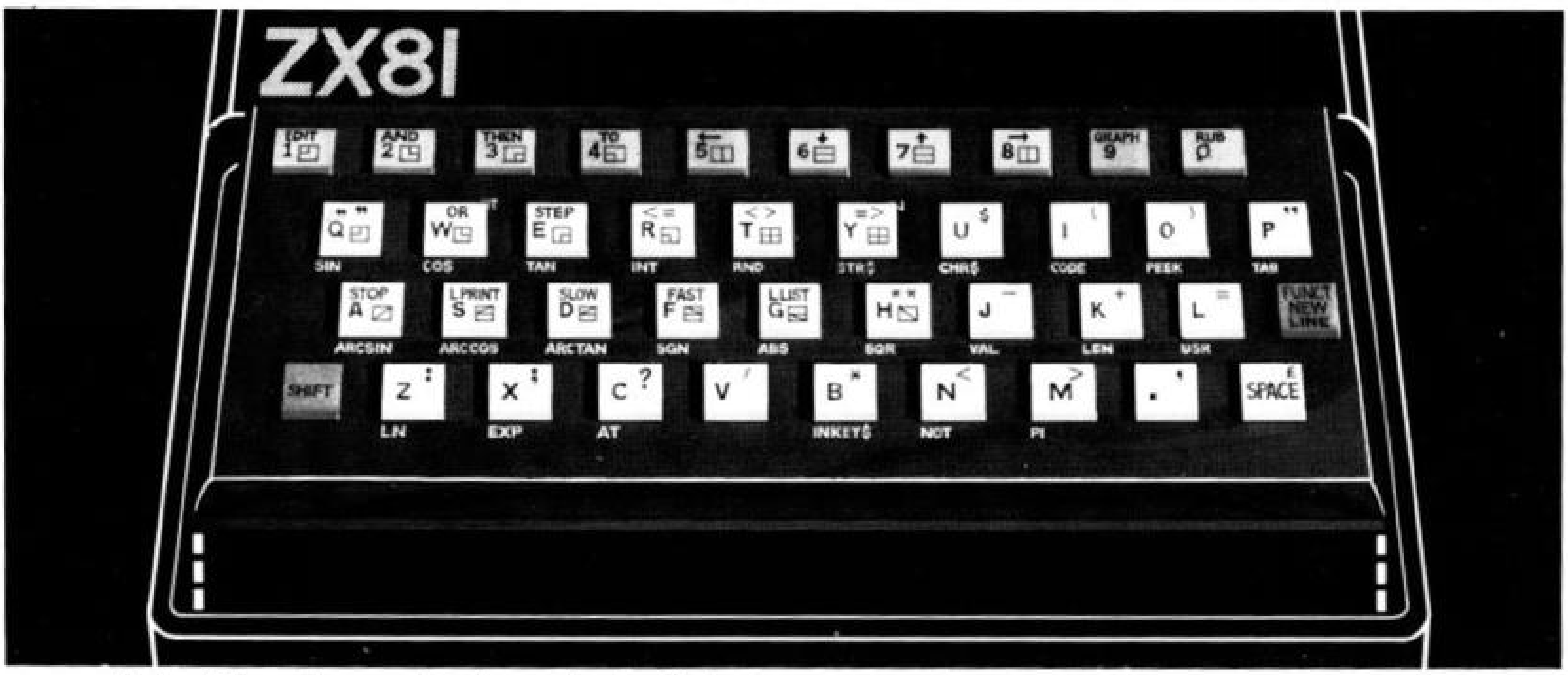
Fig. 1. One column of the screen.



```
   1 REM zzzzzzzzzzzzzzzzzzzzzzz
zzzzzzzzz
   2 REM zzzzzzzzzzzzzzzzzzzzzzz
zzzzzzzzz
   3 REM zzzzzzzzzzzzzzzzzzzzzzz
zzzzzzzzzzz
   4 REM zzzzzzzzzzzzzzzzzzzzzzz
zzzzzzzzz
   5 REM zzzzzzzzzzzzzzzzzzzzzzz
zzzzzzzzz
   6 RESTORE 6: FOR g=0 TO 28: R
EAD h: POKE 23760+g,h: NEXT g
   7 FOR g=0 TO 28: READ h: POKE
 23798+g,h: NEXT g
   8 FOR g=0 TO 33: READ h: POKE
 23836+g,h: NEXT g
   9 FOR g=0 TO 25: READ h: POKE
 23876+g,h: NEXT g
  10 FOR G=0 TO 9: READ A: POKE
23914+G,A: NEXT G
  11 RESTORE 2000: FOR H=0 TO 31
: FOR g=0 TO 21
  12 IF g=0 OR g=8 OR g=16 THEN
READ a: POKE 23915,a
  13 LET x=g-(g>=8)*8-(g>=16)*8
  14 PRINT AT g,h;: POKE 23917,1
27-x*8
  15 RANDOMIZE USR 23914
  16 NEXT g: NEXT h
  20 GO TO 500
 100 DATA 13,17,0,88,33,1,88,1,2
55,2
 110 DATA 237,176,33,31,88,14,24
 120 DATA 58,141,92,17,32,0,119,
25
 130 DATA 13,200,24,250
 200 DATA 13,17,255,90,33,254,90
,1,255,2
 210 DATA 237,184,33,0,88,14,24
 220 DATA 58,141,92,17,32,0,119,
25
 230 DATA 13,200,24,250
 300 DATA 6,8,33,31,64,22,0,30
 310 DATA 65,14,32,175,126,23
 320 DATA 119,43,13,32,249,25,62
 330 DATA 88,188,40,3,43,24,235
 340 DATA 5,175,184,32,225,201
 400 DATA 6,8,33,0,64,14,32,175
 410 DATA 126,31,119,35,13,32,24
9
```

```
420 DATA 62,88,188,32,241,5,175
430 DATA 184,32,233,201
450 DATA 62,0,203,127,40,3,62
460 DATA 143,215,201
500 LET Y=0: : LET x=11
510 POKE 23672,0: POKE 23673,0:
 POKE 23674,0
520 DEF FN a()=INT ((PEEK 23672
+PEEK 23673*256+PEEK 23674*65535
)/50)
580 IF y=107 THEN PRINT AT 10,1
5;"CONGRATULATIONS!";AT 12,16;"Y
OU COMPLETED";AT 14,17;"THE MAZE
 IN ";AT 16,18;
590 IF y=107 THEN PRINT INT ((P
EEK 23672+PEEK 23673*256+PEEK 23
674*65535)/50);" SECONDS.": STOP
600 PRINT AT x,10;"A"
610 LET a$=INKEY$
620 IF a$="5" AND POINT (74,175
-8*x)<>1 THEN GO SUB 8500
630 IF a$="8" AND POINT (90,175
-8*x)<>1 THEN GO SUB 8000
640 IF a$="7" AND POINT (82,183
-8*X)<>1 THEN PRINT AT x,10;" ":
 LET x=x-1: GO TO 600
650 IF a$="6" AND POINT (82,167
-8*X)<>1 THEN PRINT AT x,10;" ":
 LET x=x+1: GO TO 600
660 GO TO 580
1999 DATA 0,0,0
2000 DATA 255,255,255
2001 DATA 128,16,4
2002 DATA 191,119,244
2003 DATA 132,2,4
2004 DATA 245,250,252
2005 DATA 132,8,4
2006 DATA 181,239,247
2007 DATA 181,128,4
2008 DATA 183,191,215
2009 DATA 176,128,20
2010 DATA 158,239,244
2011 DATA 200,192,20
2012 DATA 227,255,212
2013 DATA 174,0,20
2014 DATA 136,223,244
2015 DATA 235,130,20
2016 DATA 138,56,84
2017 DATA 186,239,212
2018 DATA 162,136,20
2019 DATA 186,190,244
2020 DATA 138,32,132
2021 DATA 235,251,188
2022 DATA 175,0,132
2023 DATA 161,126,244
2024 DATA 189,98,20
2025 DATA 161,75,212
2026 DATA 173,88,84
2027 DATA 173,19,116
2028 DATA 165,254,84
2029 DATA 164,32,68
2030 DATA 167,247,220
2031 DATA 176,132,4
2032 DATA 190,191,244
2033 DATA 162,130,36
2034 DATA 184,184,180
2035 DATA 141,143,148
2036 DATA 229,32,212
2037 DATA 133,62,20
2038 DATA 189,227,212
2039 DATA 160,136,20
2040 DATA 190,175,244
2041 DATA 128,130,4
2042 DATA 238,250,244
2043 DATA 138,16,4
2044 DATA 187,215,252
2045 DATA 160,80,4
2046 DATA 173,87,244
2047 DATA 165,16,132
2048 DATA 181,118,188
2049 DATA 161,54,188
2050 DATA 140,22,188
2051 DATA 231,246,188
2052 DATA 178,38,132
2053 DATA 178,160,180
2054 DATA 134,190,180
2055 DATA 180,130,180
2056 DATA 145,190,180
2057 DATA 223,0,52
2058 DATA 136,127,164
2059 DATA 187,0,172
2060 DATA 169,255,172
2061 DATA 172,192,44
2062 DATA 166,94,236
2063 DATA 176,194,140
2064 DATA 163,154,188
2065 DATA 186,186,132
2066 DATA 162,10,244
2067 DATA 170,250,4
2068 DATA 170,67,244
2069 DATA 139,26,4
2070 DATA 187,250,252
2071 DATA 162,2,4
2072 DATA 174,187,180
2073 DATA 160,162,36
2074 DATA 191,175,116
2075 DATA 188,40,68
2076 DATA 161,227,220
2077 DATA 175,8,68
2078 DATA 162,123,244
2079 DATA 184,216,132
2080 DATA 142,139,188
2081 DATA 162,169,132
2082 DATA 250,43,52
2083 DATA 139,234,116
2084 DATA 160,72,196
2085 DATA 191,94,92
2086 DATA 160,67,76
2087 DATA 175,121,100
2088 DATA 161,109,52
2089 DATA 189,73,148
2090 DATA 161,91,212
2091 DATA 175,66,20
2092 DATA 165,91,116
2093 DATA 177,73,68
2094 DATA 189,109,92
2095 DATA 169,9,68
2096 DATA 171,227,84
2097 DATA 160,63,84
2098 DATA 190,224,84
2099 DATA 168,175,244
2100 DATA 171,162,36
2101 DATA 168,58,172
2102 DATA 175,162,36
2103 DATA 160,174,180
2104 DATA 190,160,164
2105 DATA 130,191,180
2106 DATA 186,160,4
2107 DATA 186,183,188
2108 DATA 162,160,4
2109 DATA 190,191,116
2110 DATA 128,132,68
2111 DATA 255,245,220
2112 DATA 128,4,68
2113 DATA 191,223,212
2114 DATA 164,80,20
2115 DATA 173,123,244
2116 DATA 160,0,4
2117 DATA 191,255,252
7999 DATA 0,0,0
8000 PRINT AT X,10;" "
8010 RANDOMIZE USR 23760: RANDOM
IZE USR 23836
8020 PRINT AT X,10;"A"
8030 LET Y=Y+1: RESTORE 2031+Y:
FOR G=0 TO 21: IF G=0 OR G=8 OR
G=16 THEN READ A: POKE 23915,A
8032 LET J=G-(G>=8)*8-(G>=16)*8
8034 POKE 23917,127-J*8
8035 PRINT AT G,31;
8036 RANDOMIZE USR 23914: NEXT G
: RETURN
8500 PRINT AT X,10;" "
8510 RANDOMIZE USR 23798: RANDOM
IZE USR 23876
8520 PRINT AT X,10;"A"
8530 LET Y=Y-1: RESTORE 2000+Y:
FOR G=0 TO 21: IF G=0 OR G=8 OR
G=16 THEN READ A: POKE 23915,A
8532 LET J=G-(G>=8)*8-(G>=16)*8
8534 POKE 23917,127-J*8
8535 PRINT AT G,0;
8536 RANDOMIZE USR 23914: NEXT G
: RETURN
```

NOW. A ZX81 PUSH-BUTTON KEYBOARD FOR UNDER £10.



At last there's a really cheap but efficient way of ironing out the ZX81's only real bug: its keyboard. The Filesixty Buttonset offers

- A full-travel calculator-type moving keyboard for only £9.95.
- Installed in seconds. The peel-off adhesive backing means you just register into position and press.
- No messy labels, dismantling or soldering.
- 3 groups of colour keys to pick out shift, numerals and newline.
- Precision moulded in ABS to match your ZX81, with contrasting legends for maximum legibility.

Allow up to 28 days for delivery.
Orders to Filesixty Ltd., 25 Chippenham Mews, London W9 2AN.

Cheques/PO made payable to Filesixty Ltd.

Please send me ______ (qty.) Buttonset(s) at £9.95 each (including VAT and P&P).

Total £ ______ BLOCK CAPITALS

Name ______

Address ______

ZXC1

FILESIXTY

Educational software

For those of you looking for some good educational software, David Valentine airs his views on the subject and reaches a very worthwhile conclusion.



To anyone who believes in computers and who understands the impact that computers are going to have on the lives of nearly everyone, it might come as something of a surprise to find that the vast majority of teachers are unaware of what computers can do to assist and enrich the learning situation. It would probably come as an even greater surprise to find that there is active resistance to the introduction of computers into the classroom. Even in some of those schools where computers have been purchased, they have been put into a separate room with restricted access to them, and are 'taught' as a separate subject, just like 'Geography' or 'Science'.

There are two questions which need answering here. One is 'Why is there such a reluctance on the part of schools to purchase computers on any scale?', and the second is 'Why are computers so badly used in schools that have purchased them?'.

With the coming of the ZX range of computers, cost can no longer be cited as the prime reason for not purchasing computers. Now that a ZX81 plus a 16K RAM pack can be purchased for as little as £75, it is possible for a school to obtain eight for the cost of one of the more usual school computers.

Catch 22

I am convinced that the answer to both questions is tied up with a general lack of suitable educational software. There are many programs on the market which are supposed to be educational. A number of companies invite respectability by tagging some pseudo-educational programs on to the end of their ranges. These programs are quite easy to spot as they are invariably of the 'quiz' type with the computer posing test questions and the child keying in answers usually from a multiple choice list. These programs are not educational; they may be fun to use once or twice, but they usually prove to be of little use for a teacher or parent to use on a regular basis.

Why is it then that there is precious little educational software available at present? There are a number of reasons. One is that there is very little profit to be made from writing such software. Whilst 'space invader' type programs sell in their thousands, even the most successful maths packages sell in only hundreds! The main problem is, however, a lack of suitable software authors. The best people to write such software are, presumably, teachers. Teachers, however, tend to be rather a conservative set of people and with teachers unsure of using them in the first place, we have a 'catch 22' type situation.

This is what they want

What then makes a good educational software package? Firstly it must be easy to use and understand. It is simply a waste of time to produce a program that makes superb use of a computer's graphic capabilities or uses some very clever programming technique if the user cannot understand how to operate it. Secondly, the program should teach or test a topic at least as well as existing methods. Computers will remain underused, gathering dust in stock cupboards, if they cannot significantly improve on a teacher's skill at the blackboard. Thirdly, the programs should be well error trapped. It is not good enough for a program to crash out when an inexperienced user is working with it. It is the same as when a film snaps in a projector — it totally disrupts the flow of the lesson and valuable time is lost. Unreliable technology, whether hardware or software, is soon consigned to the scrapheap in schools!

Fourthly, the programs should be well documented. This should include such simple information as to how to load the program, what the program is about and what age range it is intended for. It should also indicate what options are open to the user and wherever possible, how the program could be adapted to be used in different situations. It should go without saying that the program should be stimulating and visually well-presented. When, and only when, there are large banks of software dealing with all subject areas, dealing with many areas within a subject, to teach, test, illustrate and record, will computers emerge from the safety of their own suites and become commonplace in the ordinary classroom.

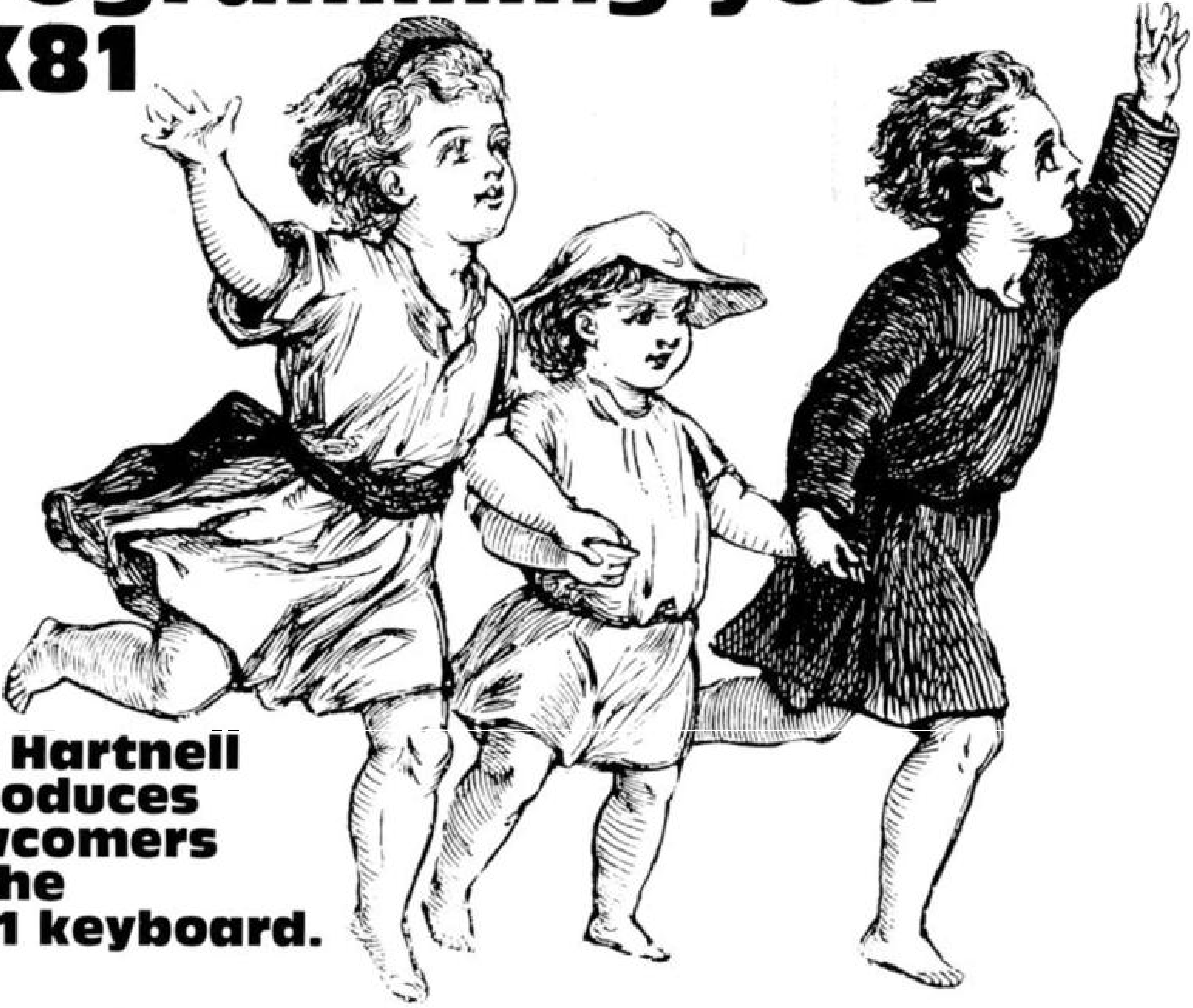
Look and learn

How can teachers and parents find what is available at present and how suitable the available software is? Now that there are an increasing number of shops who specialise in home computers, with the emphasis on the 'ZX' range, such as Microware of Leicester or the Buffer Shop in London, it ought to be possible to see educational programs demonstrated. Another method of ascertaining the suitability of software is to study the impartial and thorough reviews in computing publications.

There are a few software companies who specialise in the production of educational software. The ones who advertise regularly and who have received good reviews of their software are worth contacting for a catalogue. Also, E.Z.U.G., the Educational ZX Users Group, lay down rigorous standards for the acceptance of programs into its library, and is therefore an obvious place to search for available programs.

For those who are dissatisfied with the range and quality of the currently available educational software, there is only one course of action open — *get writing!*

First steps in programming your ZX81



Tim Hartnell introduces newcomers to the ZX81 keyboard.

The first time you look at the ZX81 keyboard, you are likely to be in for a bit of a shock. As well as numbers and letters more or less where you would expect them to be on the typewriter keyboard, there seems to be a bewildering collection of odd symbols and words within the key outline, with other words above and below the keys. Trying to work out how to get what you want from a key — and some keys can produce as many as five different results — can seem very difficult.

But it is not. The computer is designed so that it knows, more often than not, which of the five possibilities you will need. And when the computer cannot tell from the context of what you're typing in which part of the key's possibilities you want, it is very easy to instruct it.

Plug in your computer as shown in the manual which came with it. An inverse K (a white K on a little black square) will appear in the bottom left-hand corner. This is called the cursor and it is the key to working out which possibility you'll get when you press a key. If the cursor is a K, you'll get numbers or 'keywords' (the words in white above the keys). We'll look at keywords and the other possibilities in more detail shortly. If the cursor is an inverse L, the keyboard works more or less like a typewriter. That is, you press the S key, and the letter S appears on the screen. The other two possibilities for the cursor are an F (function mode) or a G (graphics mode).

The Shift key (the one with the word SHIFT in red, in the bottom left-hand corner of the keyboard) allows you to get the words and symbols written in red on the keys, as you'll see in a moment.

Slow/Fast

When you first turn the computer on, it is in Slow mode. In Slow mode, looking after the smoothness of the television picture is considered more important than 'thinking', so the computer does its thinking between sending picture information to the television. The great majority of ZX81 programs, like nearly all those in this issue of *ZX Computing*, are designed to be run in Slow mode, so the computer is in the correct mode for running them automatically. In Fast mode, the computer's thinking is considered more important than looking after the picture, so it can become very jerky.

However, it is much easier to enter a program into the ZX81 when it is in Fast mode, so we need to be able to switch between modes at will — Fast to get a program into the computer and Slow for RUNning it.

As I said, the computer is in Slow mode automatically when you turn it on. To get it into Fast mode, press on the Shift key (remember, it's the bottom left-hand corner of the keyboard) then press the F key (which, as you'll see, has the word FAST written on it in red). The word FAST should appear on the screen. Now press the Enter/Newline key (second from the bottom, on the right-

hand side), and you'll see the screen jump, and the 'message' 0/0 appear in the bottom left-hand corner of the screen. This message — 0/0 — means 'all OK'.

Now press any key, and you'll see the display jumping about. This is what Fast mode looks like; far less attractive to watch, but far easier to use for program entry.

You may find when you write your own programs that the speed of Slow is, in fact, too slow for your liking. This can happen when the computer must do a great deal of work to determine an answer, and in this sort of program — when the display is less important than getting a quick answer — you should run your programs in Fast. However, for many programs (such as the majority of those in this magazine), the ZX81 is still fast enough in Slow for our needs.

To get back into Slow mode from Fast, hold down the Shift key again, then press the D key (where the word SLOW is written in red). The display will 'lock on', and the 'all OK' message will appear on the screen. From now on, we'll assume that you have your computer in Fast mode when you enter programs, and in Slow mode when you RUN them.

Keywords

The words written in white *above* many of the keys are 'keywords'. These are the fundamental words from which programs are built — the main building blocks of your programs. When you write a program such as you see throughout this issue of *ZX Computing,* you start with a number (the line number), and follow this with a keyword, then generally some additional material.

Here's an example which should help to make that last sentence clear. Make sure your computer is 'empty', which you can do by turning off the power, waiting a few seconds, then turning it on again. This is not the best way to clear the contents of the computer's memory (the word NEW above the A key is designed to do it), but this method is the simplest at this stage.

Once the computer is empty, type in the number 10. It will sit down at the bottom of the screen. The 10 is a line number. Now press the P key, and you'll see the word PRINT appear on the screen, just after the 10. The keywords appear automatically after a line number. Now press Enter (it may be called Newline on your ZX81, as this was used for this key on earlier models), and you'll see the program line move up to the top of the screen, which means it has been accepted by the computer.

Functions

Functions are the words in white *under* many of the keys. The functions include such esoteric words as LEN, INKEY$ and TAB. You get into the function mode (when the inverse K cursor turns into an inverse F) by holding down the Shift key, then pressing the Enter key, which you can see has the word FUNCTION on it in red, above the word ENTER (or NEWLINE). When you do this, you'll see the cursor has turned into an inverse F. Now press the C key, and the word AT should appear on the screen. Try the B key (to get the word INKEY$) and the P key (to get TAB).

Operators

The operators are the link and comparison words and symbols like THEN, AND, TO and < > . You get these by holding down the Shift key, then pressing the key the operator you want is on. Clear the computer as before, then hold down the Shift key and press U. The dollar sign should appear. Press 2 and you'll get the word AND, 3 for THEN and 4 for TO.

The word in red on the 1 key — EDIT — has a special function. It is used to bring a line down from the top of the screen (where it has been accepted into the computer's memory as part of a program) to the bottom of the screen so you can change or EDIT it in some way.

To show this in action, clear the computer, then type in the following:

```
10 PRINT 6
```

Press Enter and this line will move to the top of the screen. Now hold down the Shift key, and press the 1 key, and the line will reappear at the bottom of the screen. Still holding down the Shift, press the 8 key (where you'll see a little arrow pointing to the right) and you'll see the cursor move across the word PRINT to rest between it and the 6. Press the 8 key again (still holding down Shift) and the cursor will jump over the word 6. The word DELETE (or RUBOUT) is on the 0 key (and note that the zero has a line through it to distinguish it from the letter O). Still holding down Shift, press the 0 key once, and the number 6 will be erased. Take your finger off the Shift and press the 5 key, to see the number 5 appear at the end of the line. What you have (or what you should have) at the bottom of the screen will look like this:

```
10 PRINT 5
```

Now press Enter and the line will appear at the top of the screen, taking the place of the previous one.

Graphics

The final cursor mode we'll discuss is the graphics mode.

Clear the computer's memory with New (Shift, then press the A key so the word NEW appears, then press Enter). Now type in the number 10, then press the P key, so the keyword PRINT appears. You should now have this at the bottom of the screen:

```
10 PRINT
```

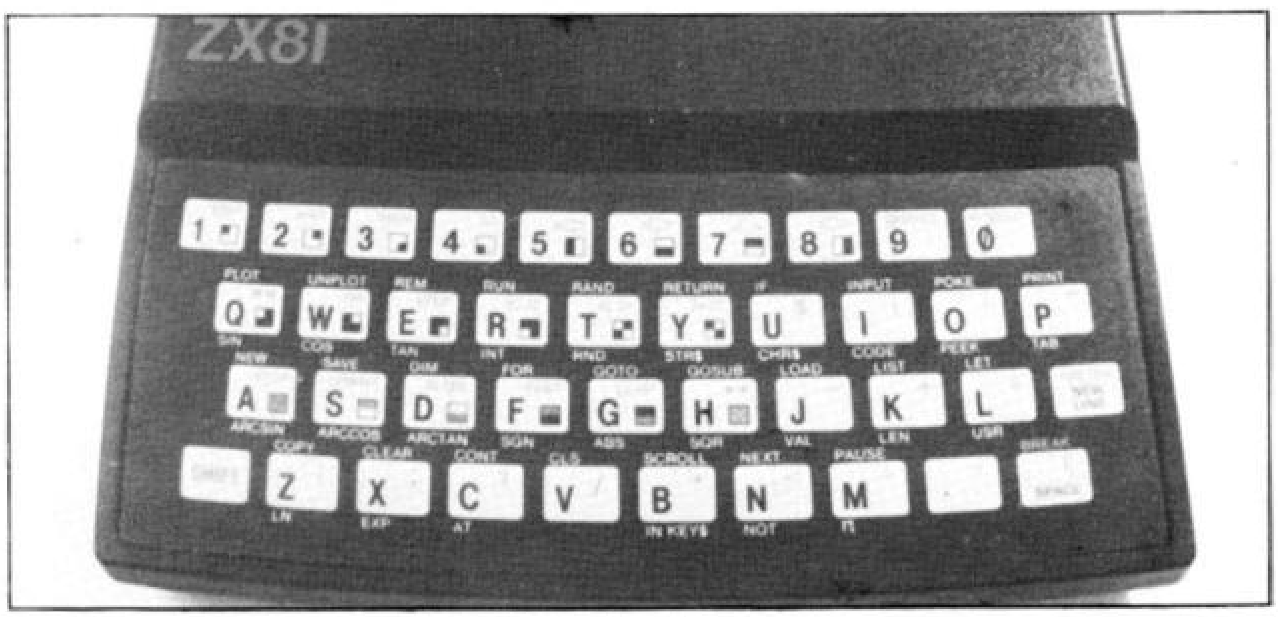
Now hold down the Shift key, and press the P key again, and a pair of quote marks will appear. Still holding down Shift, press the 9 key, and the cursor will turn into a G, meaning the computer is in the graphics mode. Now type in a word, and you'll see it appear in white letters on little black squares, rather than in black letters on the ordinary white background. Holding down the Shift again, press the 9 key again, and the inverse G will turn back into an L. Still holding down Shift, press the P key for the closing quote marks. Now press Enter, and your line (complete with a word in inverse letters) should appear at the top of the screen.

RUN this program (by pressing the R key to get the word RUN at the bottom of the screen, then pressing the Enter key) and your word, in white letters on a little black background strip will appear at the top of the screen.

If you are in the graphics mode, and you hold down the Shift key while pressing some of the keys, instead of an inverse letter you'll get the little patterns and designs on the keys. These can be used to build up pictures.

All keyed up

Now, I don't expect you'll understand all of this discussion instantly, and hope you'll refer back to this article from time to time as you continue working your way through this magazine. You should, however, find you quickly master the fundamentals of the keyboard, and the need to refer to this introductory article will diminish as time goes by. You may wish to read through the whole of this article again now, before you tackle the programs on other pages.



ADVERTISEMENT INDEX

MACHINE SPECIFICATIONS

ZX80

Dimensions
Width 174mm (6.85 in)
Depth 218mm (8.58 in)
Height 38 mm (1. 5 in)
Weight 300g (10.5oz)

Microprocessor/Memory
Z80A 3.25 MHz clock
ROM: 4K bytes containing BASIC
RAM: 1K bytes internal, externally expandable to 16K bytes.

Display
Requires an ordinary domestic black and white colour TV. The lead supplied connects between the ZX80 and your TV's aerial socket. The display organisation is 24 lines of 32 characters per line showing black characters on a white screen. The ZX80 does not connect to a printer.
Programming
Programs can be entered on the keyboard or loaded from cassette. The ZX80 has automatic "wrap round" so lines of program can be any length but not multi-statement lines.
Syntax check
The syntax of the entered line is checked character by character. A syntax error cursor marks the first place the syntax breaks down if there is an error. Once any errors have been edited out the syntax error cursor disappears. Only syntax error-free lines of code are accepted by the ZX80.
Graphics
Total of 22 graphics symbols giving 48 x 64 pixels resolution consisting of 10 symbols plus space and inverses. Includes symbols for drawing bar charts. Under control of your BASIC program any character can be printed in reverse field.
Editing
The line edit allows you to edit any line of program or input including statement numbers. The edit and cursor control keys are EDIT, RUBOUT, HOME.
Arithmetic
Arithmetic operators +,–,x, ÷ exponentiate. Relational operators <, >, = , yielding 0 or – 1. Logical operators AND OR NOT yielding boolean result. Relational operators also apply to strings. ZX80 BASIC uses 16 bit two's complement arithmetic (± 32767).
Variables
Numeric variable names may be any length, must begin with a letter and consist of alphanumerics. Every character in the name is compared thus an infinity of unique names is available.
String variables may be assigned to or from, shortened but not concatenated. String variable names are A$ – Z$. Strings do not require a dimension statement and can be any length.
Arrays have a maximum dimension of 255 (256 elements) each. Array names consist of a single letter A–Z.
Control variable names in FOR. . . NEXT loops consist of a single letter A–Z.

Expression evaluator
The full expression evaluator is called whenever a constant or variable is encountered during program execution. This allows you to use expressions in place of constants especially useful in GOTOs, GOSUBs, FOR. . . NEXT etc.

Immediate mode
The ZX80 will function in the "calculator mode" by immediately executing a statement if it is not preceded with a line number.
Cassette interface
Works with most domestic cassette recorders. The transfer rate is 250 baud using a unique tape-recording format. Other systems are not compatible with the ZX80's. The ZX80 also SAVEs the variables as well as the program on cassette. Therefore you can save the data for updating next time the program is executed. The ZX80 does not support separate data files. The lead supplied with the ZX80 is fitted with 3.5mm jack plugs.
Expansion bus
At the rear has 8 data, 16 address, 13 control lines from the processor and Ov, 5v, 9-11v, Ø and internal memory control line. These signals enable you to interface the ZX80 to your own electronics, PIO, CTC, SIO if you want I/O ports etc.
Power supply
The ZX80 requires approximately 400mA from 7–11v DC. It has its own internal 5v regulator.
TV standard
The ZX80 is designed to work with UHF TVs (channel 36)and is the version required for use in the United Kingdom. The ZX80 USA is designed to work with a VHF TV(American channel 2. European channel 3) and is the version required for the American TV system, also for countries without UHF.

ZX81

Dimensions
Width 167mm (6.32 in)
Depth 175mm (6.80 in)
Height 40 mm (1.57 in)
Weight 350 gms (12.15 oz)

Microprocessor/Memory
Z80A 3.25 MHz clock
ROM: Containing 8K BASIC interpreter
RAM: 1K bytes internal, externally expandable to 16K bytes.

Keyboard
40 key touch-sensitive membrane. Using function mode and single press key-word system, this gives the equivalent of 91 keys and also graphics mode allows an additional 20 graphical and 54 inverse video characters to be entered directly.
Display
Requires an ordinary domestic black and white or colour TV. The aerial lead supplied connects the ZX81 to the TV aerial socket. The display is organised as 24 lines of 32 characters with black characters on a white background.
Two mode speeds
The ZX81 can operate in two software-selectable modes - FAST and NORMAL. FAST is ideal for really high-speed computing. In NORMAL mode however the ZX81 allows continuously moving, flicker-free animated displays
Printer
The 8K ROM will permit instructions (LPRINT, LLIST and COPY) to drive the Sinclair ZX Printer.
Programming
Programs can be entered via the keyboard or loaded from cassette. Programs and data can be saved onto cassette so that they

are not lost when the ZX81 is turned off.

Syntax check

The syntax of a line of program is checked on entry. A syntax error cursor marks the first place the syntax breaks down if there is an error. The syntax error cursor disappears when errors have been corrected. Only lines free from syntax errors will be entered into the program.

Graphics

Apart from the 20 graphics characters, space and its inverse, the display may also be divided into 64 x 44 pixels, each of which may be 'blacked' in or 'whited' out under program control.

Editing

A line editor allows you to edit any line of program or input, including program line numbers. Lines may be deleted, increased or decreased in size.

Arithmetic

Arithmetic operators +, –, x, ÷, exponentiate. Relational operators =, < >, >, <,< =,>=, may compare string and arithmetic variables to yeild 0 (False) or 1(True). Logical operators AND, OR, NOT yield boolean results.

Floating-point numbers

Numbers are stored in 5 bytes in floating-point binary form giving a range of $\pm 3 \times 10^{-39}$ to $\pm 7 \times 10^{38}$ accurate to 9½ decimal digits.

Scientific functions

Natural logs/antilogs; SIN, COS, TAN and their inverses;SQR; e^x.

Variables

Numerical:	any letter followed by alphanumerics
String:	A$ to Z$
FOR-NEXT loops:	A–Z (loops may be nested to any depth.
Numerical arrays:	A–Z
String arrays:	A$ to Z$

Arrays

Arrays may be multi-dimensional with subscripts starting at 1.

Expression evaluator

The full expression evaluator is called whenever an expression, constant or variable is encountered during program execution. This powerful feature allows use of expressions in place of constants and is especially useful in GOTO, GOSUB etc.

Command mode

The ZX81 will execute statements immediately, enabling it to perform like a calculator.

Cassette interface

Works using domestic cassette recorders. The transfer rate is 250 baud and uses a unique recording format not compatible with other systems. The ZX81 will save the data as well as the program to avoid the need to re-enter the data when the program is next loaded.,

ZX81 will search through a tape for the required program).

The cassette leads supplied have 3.5 mm jack plugs.

Expansion port

At the rear, this has the full data, address and control buses from the Z80A CPU as well as OV, +5V, +9V,Ø and the memory select lines. These signals enable you to interface the ZX81 to the Sinclair 16K RAM pack and ZX printer.

Power supply

The ZX81 requires approximately 420mA at 7–11V DC. It has its own internal 5V regulator. The ready assembled ZX81 comes complete with a power supply. The ZX81 kit does not include a power supply.

TV standard

The ZX81 is designed to work with UHF TVs (channel 36) 625 lines.

ZX SPECTRUM

Dimensions

Width 233 mm
Depth 144 mm
Height 30 mm

CPU/Memory

Z80A microprocessor running at 3.5 MHz. 16K-byte ROM containing BASIC interpreter and operating system.

16K-byte RAM (plus optional 32K-byte RAM on internal expansion board) or 48K-byte RAM.

Keyboard

40-key keyboard with upper and lower case with capitals lock feature. All BASIC words obtained by single keys, plus 16 graphics characters, 22 colour control codes and 21 user-definable graphics characters. All keys have auto repeat.

Display

Memory-mapped display of 256 pixels x 192 pixels; plus one attributes byte per character square, defining one of eight foreground colours, one of eight background colours, normal or extra brightness and flashing or steady. Screen border colour also settable to one of eight colours. Will drive a PAL UHF colour TV set, or black and white set (which will give a scale of grey), on channel 36.

Sound

Internal loudspeaker can be operated over more than 10 octaves (actually 130 semitones) via basic BEEP command. Jack sockets at the rear of computer allow connections to external amplifier/ speaker.

Graphics

Point, line, circle and arc drawing commands in high-resolution graphics.

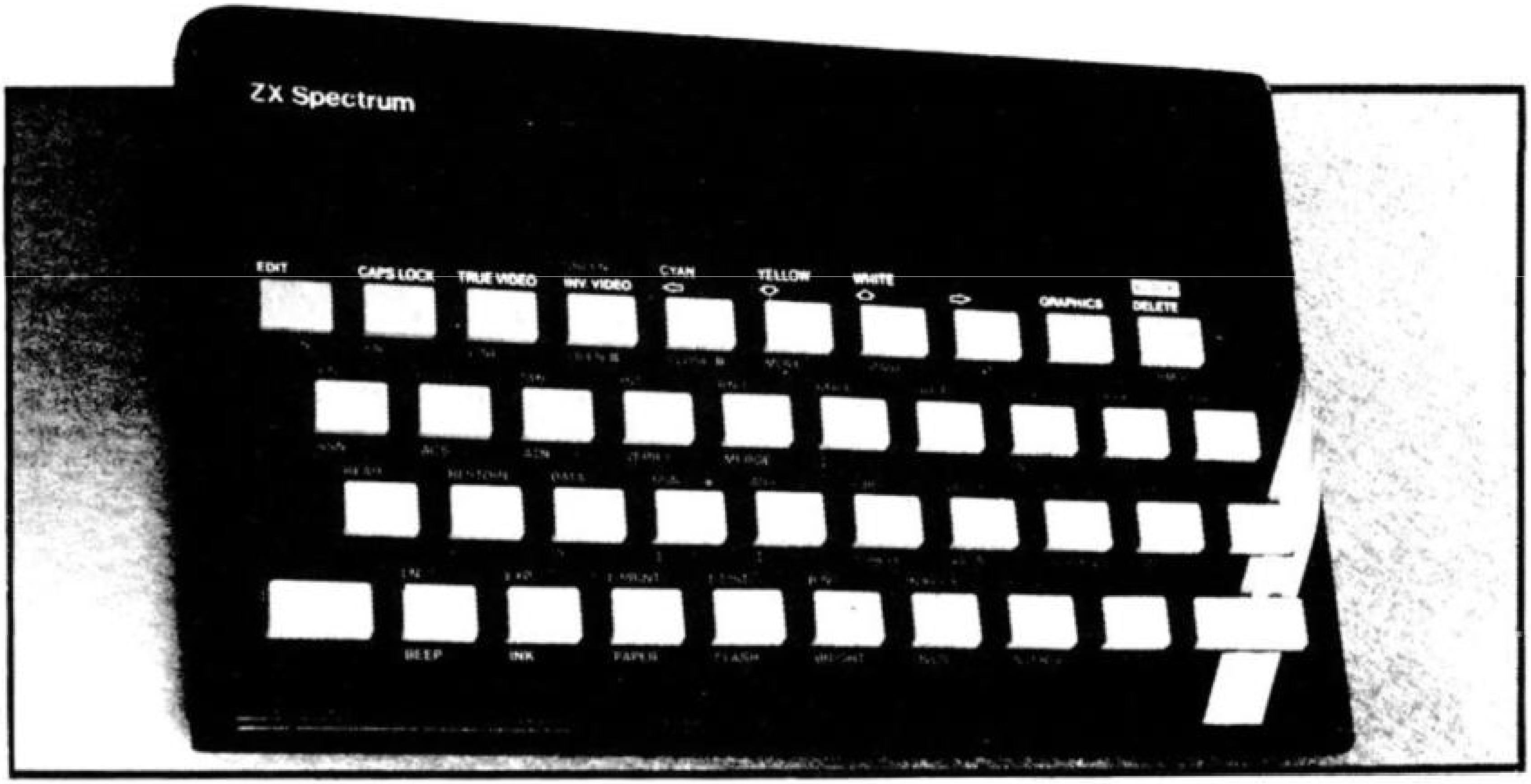
16 pre-defined graphics characters plus 21 user-definable graphics characters. Also functions to yield character at a given position, attribute at a given position (colours, brightness and flash) and whether a given pixel is set. Text may be written on the screen on 24 lines of 32 characters. Text and graphics may be freely mixed.

Colours

Foreground and background colours, brightness and flashing are set by BASIC INK, PAPER, BRIGHT and FLASH commands. OVER may also be set, which performs an exclusive – or operation to overwrite any printing or plotting that is already on the screen. INVERSE will give inverse video printing. These six commands may be set globally to cover all further PRINT, PLOT, DRAW or CIRCLE commands, or locally within these commands to cover only the results of that command. They may also be set locally to cover text printed by an INPUT statement. Colour-control codes, which may be accessed from the keyboard, may be inserted into text or program listing, and when displayed will override the globally set colours until another control code is encountered. Brightness and flashing codes may be inserted into program or text, similarly. Colour-control codes in a program listing have no effect on its execution. Border colour is set by a BORDER command. The eight colours available are black, blue, red, magneta, green, cyan, yellow and white. All eight colours may be present on the screen at once, with some areas flashing and others steady, and any area may be highlighted extra bright.

Screen

The screen is divided into two sections. The top section – normally the first 22 lines – displays the program listing or the results of program or command execution. The bottom section – normally the last 2 lines – shows the command or program line currently being entered, or the program line currently being edited. It also shows the report messages. Full editing facilities of cursor left, cursor right, insert and delete (with auto-repeat facility) are available over this line. The bottom section will expand to accept a current line of up to 22 lines.



Mathematical Operations And Functions

Arithmetic operations of +, −, ×, ÷, and raise to a power. Mathematical functions of sine, cosine, tangent and their inverses; natural logs and exponentials; sign function, absolute value function, and integer function; square root function, random number generation, and pi.
Numbers are stored as five bytes of floating point binary — giving a range of $+3 \times 10^{-39}$ to $+7 \times 10^{38}$ accurate to 9½ decimal digits.
Binary numbers may be entered directly with the BIN function. =, >, <, > =, < = and < > may be used to compare string or arithmetic values or variables to yield 0 (false) or 1 (true). Logical operators AND, OR and NOT yield boolean results but will accept 0 (false) and any number (true).
User-definable functions are defined using DEF FN, and called using FN. They may take up to 26 numeric and 26 string arguments, and may yield string or numeric results.
There is a full DATA mechanism, using the commands READ, DATA and RESTORE.
A real-time clock is obtainable.

String Operations And Functions

Strings can be concatenated with +. String variables or values may be compared with =, >, <, > =, < =, < > to give boolean results. String functions are VAL, VAL$, STR$ and LEN. CHR$ and CODE convert numbers to characters and vice versa, using the ASCII code. A string slicing mechanism exists, using the form a$ (x TO y).

Variable Names

Numeric — any string starting with a letter (upper and lower case are not distinguished between, and spaces are ignored).
String — A$ to Z$.
FOR-NEXT loops — A-Z.
Numeric arrays — A-Z.
String arrays — A$ to Z$.
Simple variables and arrays with the same name are allowed and distinguished between.

Arrays

Arrays may be multi-dimensional, with subscripts starting at 1. String arrays, technically character arrays, may have their last subscript omitted, yielding a string.

Expression Evaluator

A full expression evaluator is called during program execution whenever an expression, constant or variable is encountered. This allows the use of expressions as arguments to GOTO, GOSUB, etc.
It also operates on commands allowing the ZX Spectrum to operate as a calculator.

Cassette Interface

A tone leader is recorded before the information to overcome the automatic recording level fluctuations of some tape recorders, and a Schmitt trigger is used to remove noise on playback.
All saved information is started with a header containing information as to its type, title, length and address information. Program, screens, blocks of memory, string and character arrays may all be saved separately.
Programs, blocks of memory and arrays may be verified after saving.
Programs and arrays may be merged from tape to combine them with the existing contents of memory. Where two line numbers or variables names coincide, the old one is overwritten.
Programs may be saved with a line number, where execution will start immediately on loading.
The cassette interface runs at 1500 baud, through two 3.5 mm jack plugs.

Expansion Port

This has the full data, address and control busses from the Z80A, and is used to interface to the ZX Printer, the RS232 and NET interfaces and the ZX Microdrives. IN and OUT commands give the I/O port equivalents of PEEK and POKE.

ZX81 Compatibility

ZX81 BASIC is essentially a subset of ZX Spectrum BASIC. The differences are as follows.
FAST and SLOW: the ZX Spectrum operates at the speed of the ZX81 in FAST mode with the steady display of SLOW mode, and does not include these commands.
SCROLL: the ZX Spectrum scrolls automatically, asking the operator "scroll?" every time a screen is filled.
UNPLOT: the ZX Spectrum can unplot a pixel using PLOT OVER, and thus achieves unplot.
Character set: the ZX Spectrum uses the ASCII character set, as opposed to the ZX81 non-standard set.



ADVERTISEMENT RATES

1. SEMI — DISPLAY

(min 2 cm)	£7.25 per cm
4-11 insertions	£6.75 per cm
12 + insertions	£6.25 per cm

***or* 2. LINEAGE**

30p per word. (minimum 15 words)
Box Numbers £2.50

Closing date for the next issue is 21st. 2nd. 1983.
All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request)
Send your requirements to: ADVERTISING DEPT., ZX COMPUTING, 145 CHARING CROSS ROAD, LONDON WC2H 0EE.

ZX COMPUTING — CLASSIFIED ADVERTISEMENT — ORDER FORM

1.	2.	3.
4.	5.	6.
7.	8.	9.
10.	11.	12.
13.	14.	15.

Advertise nationally in these columns to over 100,000 readers for only 30p per word (minimum charge 15 words). Simply print your message in the coupon and send with your cheque or postal order made payable to Argus Specialist Publications Ltd to:

CLASSIFIED DEPT. ZX COMPUTING
145 Charing Cross Rd, London WC2H 0EE.
Tel: 01-437 1002

Name . . . .

Address . . . .

. . . .

. . . .

Tel.No.(Day) . . . .

Please place my advert in ZX Computing for months

Please indicate number of insertions required.

"QS Produce the most ULTRA-MEGA-AMAZING Games in the entire known Universe!"



TIMEGATE

Quicksilva's 1st level two game. The best space/time adventure of 82.

TIMEGATE

High speed graphics combine with electrifying space adventure makes TIMEGATE the game of all games.

Foil the ruling aliens by travelling back through time and blasting out their home planet. Fight 3D space battles on your journey, seeking the time gates that lead to year zero and the salvation of the human race.

Features: 4D, High speed graphics, versatile scanning, status and battle computers, land & take-off from many planets, time-travel, 5 skill levels, score & rating plus free training program.

MORE NEW RELEASES!

THE CHESS PLAYER

Who is he? Dare YOU challenge him? Risk high stakes and pit your wits against the chess game with personality, speech and devastating chess skills. THE CHESS PLAYER costs less, while offering you more skill—boosting enjoyment.

Features: 6 skill levels, analysis, copies display and all moves to printer or screen, colour or B & W display, save game at any point, Personality & speech.

SPEAKEASY

Add speech or music to you programs! In response to massive demand, we have released SPEAKEASY for your own use and enjoyment.

Simply record your sound onto a cassette and play it into your Spectrum via load. The signal may then be edited and replayed on command from your Basic or M/c program.

NO EXTRA HARDWARE REQUIRED!

Edward de Bono's Classic

L-GAME

Simple rules and complex strategies make the L-GAME an essential for everyone.

Also includes outstanding and compulsive versions of Mastermind & Pontoon.

As featured on T.V. in Edward de Bono's Thinking Course.

L-GAME REACHES THE BITS OTHER GAMES DON'T TOUCH!



SPACE INTRUDERS

Features 4 alien types, random saucer points, 2nd attack wave mutant, exploding aliens, bonus base 50,000 points, left-right & fire, sound effects and colour.

METEOR STORM

3 meteor types, 2 saucers, full screen wrap-around, bonus ship every 10,000 points, amazing explosions, left-right-thrust-fire & hyperspace, sound effects and speech.

WHAT THEY SAY ABOUT OUR ZX81 GAMES:

QS SCRAMBLE—
"...amazing, fantastic!..." (P.C.W.)

QS INVADERS—
"...just like the real thing!..." (C & V.G.)

QS ASTEROIDS—
"...very good." "...addictive game..." (C & V.G.)

QS DEFENDER—
"...Better than any other arcade game I've seen..." (Sync)

"...nicest games I've played on the ZX81. All the tapes were impressively packaged..." (Which Micro & S.R.)

COME AND SEE OUR AMAZING RANGE AT:
THE MIDLAND COMPUTER FAIR (28th - 30th April)

ZX - 81 GAMES

Scramble	m/c	16k	£3.95	☐
Invaders	m/c	16k	£3.95	☐
Asteroids	m/c	16k	£3.95	☐
Defenders	m/c	16k	£3.95	☐

SPECTRUM GAMES

Speakeasy	m/c	48k	£4.95	☐
Timegate	m/c	48k	£6.95 *	☐
The Chess Player	m/c	48k	£6.95	☐
Space Intruders	m/c	16k	£4.95 *	☐
Meteor Storm	m/c	16k	£4.95 *	☐
L-Game	Basic	16k	£3.95	☐

*THESE GAMES INCLUDE SPECIAL SEASONAL DISCOUNTS, ORDER NOW AND GET FREE KEYBOARD OVERLAY!

Please send me the games as ticked.

Total cheque/P.O. enclosed ....

Name ....

Address ....

....

....

Send order to: QUICKSILVA, DEPT SPZ, 92 NORTHAM ROAD, SOUTHAMPTON, SO2 OPB